This is to certify that the

dissertation entitled

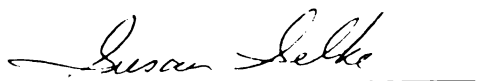
Development of a Computer Model to Evaluate
the Ability of Plastics to Act as Functional Barriers

presented by

Jong-Koo Han

has been accepted towards fulfillment
of the requirements for

Ph.D. degree in Packaging



Major professor

Date June 21, 2002

DATE DUE	DATE DUE	DATE DUE
[illegible] 2004		
NOV 8 1 7 2010		

DEVELOPMENT OF A COMPUTER MODEL TO EVALUATE THE ABILITY OF PLASTICS TO ACT AS FUNCTIONAL BARRIERS

By

Jong-Koo Han

A DISSERTATION

Submitted to
Michigan State University
In partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

School of Packaging

2002

ABSTRACT

DEVELOPMENT OF A COMPUTER MODEL TO EVALUATE
THE ABILITY OF PLASTICS TO ACT AS FUNCTIONAL BARRIERS

By

Jong-Koo Han

The overall objective of this research was the development of a computer program that can be utilized to predict the adequacy of a functional barrier structure in preventing migration of undesirable constituents of post-consumer recycled plastics into liquid products such as foods.

For this study, 3-layer co-extruded high density polyethylene (HDPE) film samples, having a symmetrical structure with a contaminated core layer and virgin outer layers as the functional barriers, were fabricated with varying thickness of the outer layers and with a known amount of selected contaminant simlulants, 3,5-di-*t*-butyl-4-hydroxytoluene (BHT) and Irganox™1076 (octadecyl-3-(3,5-di-*t*-butyl-4-hydroxyphenyl)-propionate), in the core layer. Experimentally, migration of the contaminant simulants from the core layer to the liquid food simulants was determined as a function of the thickness of the outer layer at different temperatures.

A simulation model computer program, which accounts for not only the diffusion process inside the polymer but also partitioning of the contaminant between the polymer and the contacting phase, was developed based on a numerical treatment, the finite element method, to quantify the migration through the multi-layer structures. The accuracy of the model in predicting migration was demonstrated successfully by comparing the simulated results to experimental data.

The computer program, developed as a total solution package for migration problems, can be applied not only to multi-layer structures made with the same type of plastics, but also to structures with different plastics, e.g. PP/PE/PP. This work might provide the potential for wider use of recycled plastic, especially polyolefins which have less barrier ability, in food packaging, and simplification of the task of convincing the FDA that adequate safety guarantees have been provided.

To my parents and family

ACKNOWLEDGMENTS

I would like to express most heartily thanks to my advisor, Dr. Susan Selke, for her assistance, guidance, and encouragement throughout my study.

I would also like to thank my guidance committee members, Dr. Theron Downes, Dr. Bruce Harte, and Dr. Charles Petty. They were always available for me and eager to help me in any way they can. I want to give a special thanks to Dr. Larry Segerlind for his assistance in developing the computer program.

I would like to take this opportunity to remember the late Dr. Jack Giacin. He was always supportive and friendly to me. I pray his soul may rest in peace.

I am grateful to the CFPPR at the School of Packaging for providing me financial support for this study. I would also like to thank my colleagues and the faculty at the School of Packaging for making my second graduate study so valuable and memorable. A special thanks to Youn Suk Lee.

Finally, my heart and appreciation go out to my wife and two children for their continuous support throughout the years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

A : Surface area of polymer in contact with the food

$C_{F,\theta}$: Concentration of migrant in the food at steady state or equilibrium

$C_{P,\theta}$: Concentration of migrant in the polymer at steady state or equilibrium

$C_{P,0}$: Concentration of migrant within the polymer at time 0

$C_{x,t}$: Migrant concentration in the polymer at distance x and time t

D : Diffusivity of the migrant within the polymer

K, K_L^P : Partition coefficient between the food and the polymer

L : Thickness of the polymer

R : Thickness of the core layer (recycled layer)

$M_{F,t}$: Mass that has migrated into the food after time t

$M_{F,\theta}$: Mass that has migrated into the food at steady state or equilibrium

$M_{P,\theta}$: Mass of migrant still in the polymer at steady state or equilibrium

t : Time

α : Ratio of the volumes of the food and the polymer

ρ : Density of the polymer, g/cm^3

$C_{B/2}$: Concentration of migrant at interface between the barrier layer and the food

θ_r : Lag time when the migrant contaminates the barrier layer

1. INTRODUCTION

During the past decade, pressures to utilize post-consumer recycled (PCR) plastics in packaging have increased, since use of landfills and incineration cannot be the ultimate solutions to the solid waste problem. Markets are increasingly seen as the key to permitting increased recycling and thus a decrease in waste disposal. One result in some cases is legislation that mandates recycled content in packages (Thompson Publishing Group, 2001). Plastics in direct contact with food have often been exempted, but this is, at best, likely to be temporary. Marketing considerations have also played a significant role in efforts to use recycled plastics and thus be perceived as being environmentally responsible (Thayer 1989, Powell 1995, Doyle 1996).

PCR plastics (mostly polyethylene) are already available commercially for packaging of industrial products and household cleaners; however, increased use of PCR for food packaging should be discussed and considered. Examples of special cases can already be found on the international market, e.g. polyethylene terephthalate (PET) bottles for beverages or three-layer polystyrene cups for milk products. This shows a possible trend which could lead to an even wider application of recycled plastics for food packaging in the near future (Lai and Selke 1988, Graff 1992). Recently, a 5-layer PET beer bottle containing up to 35% PCR PET was introduced commercially (Lingle 1999).

Lack of knowledge of the substances present in the recycled materials has been an obstacle for the application of recycled plastics for food packaging. Their migration potential to food needs to be examined. Polymeric packaging materials are exposed to components from the filled product and after discarding they are exposed to various external influences, as well as to additional thermal stress during waste sorting, intermediate storage, regranulating and re-extrusion to form new packaging material. As a result, there exists an inevitable potential for contamination in the recycled plastic packaging structures, either by foreign substances migrating into the plastic or by chemical reactions involving the plastic itself.

The use of functional barriers, layers of virgin resin designed to protect the food or other contents from contaminants possibly found in the recycled material, is a technique which has received "non-objection" status from the FDA in a number of cases (Ford 1994). Safety, however, must remain a paramount concern. The FDA has accepted the use of model simulants and the Begley and Hollifield diffusion model as the basis for such approvals (Begley & Hollifield, 1993). However, the model assumptions predict significantly more migration than is physically realistic. In the case of PET, the diffusivity is sufficiently low that this has not produced an absolute bar to approval of such systems. Polyolefins such as high density polyethylene (HDPE), which is one of the dominant plastic packaging materials, have less barrier ability and therefore the deficiencies of the model are a serious problem.

A model which accurately reflects the physics of the migration process from the multilayer structure could make it possible to utilize plastics with less barrier ability as a functional barrier for food packaging applications.

A simulation model based on a computer program was developed using a numerical approach, the finite element method, to quantify migration throughout the multilayer structure. The accuracy of the model in predicting migration was demonstrated by comparing the simulated results to experimental data.

Experimentally, migration of the contaminant simulant from the core layer to the liquid food simulants was determined as a function of the thickness of the outer layer at different temperatures. For the study, 3-layer co-extruded HDPE film/sheet samples, having a symmetrical structure with a contaminated core layer and virgin outer layers as the functional barriers, were fabricated with varying thickness of outer layers. As the contaminant simulants, three phenolic antioxidants were selected: BHT (3,5-di-*t*-butyl-4-hydroxytoluene), Irganox™1076 (octadecyl-3-(3,5-di-*t*-butyl-4-hydroxyphenyl)-propionate), and Irganox™1010 (pentaerythritol tetrakis (3-(3,5-di-*t*-butyl-4-hydroxyphenyl) propionate)), and ethanol and aqueous ethanol solutions were selected as the food simulants.

The major emphasis of this research was to develop a computer program (with the "Visual Basic™" programming language for "Windows™" applications) that can mathematically predict the migration process of the multilayer structure, and then to validate the computer program through

experimentation. This computer program, developed as a total solution package for migration problems, can be applied not only for multilayer structures made with the same plastics but also with different plastics, e.g. PP/PE/PP. The following are the specific objectives of the research:

1) To develop a model which accurately reflects the physics of the migration process through the multilayer structures.
2) To carry out migration studies with multilayer systems, which demonstrate and verify the accuracy and applicability of the model.
3) To develop a computer program using the model to predict performance of multilayer structures containing recycled HDPE which are intended for direct food contact.

Hypothesis

1. Migration can be accurately predicted and modeled if all the migration controlling parameters are known: the diffusion coefficient of the substance in the polymer, the mass transfer coefficient (mixing coefficient), and the partition coefficient (K) of the substance between the polymer and the food.
2. Depending on the chemical or physical affinity of the substance between the polymer and the liquid, the partition factor can differ from 0, and besides the diffusivity, partitioning is one of the most dominant factors in controlling the actual migration rate.
3. Though HDPE, which is a polyolefin, has poorer barrier properties than PET, HDPE can function as a "functional barrier".

2. LITERATURE REVIEW

2.1. Recycle of Plastics in Food Packaging

In the United States, approximately 22 percent (excluding composting) of municipal solid waste (MSW) and 37.2 percent of all "containers and packaging" was recycled in 1999. Recycling rates rose from 14.2% in 1990 to 22.1% in 1999 (EPA 2001). The main reasons for the increase in recycling are environmental regulations, consumer participation, and limitations on landfills and incineration. As economic growth results in more products and materials being generated, there will be an increased need to invest in source reduction activities such as light weighting of products and packaging. However, a more important approach will be utilizing existing recycling facilities, further developing this infrastructure, research and development of recycling techniques, and buying recycled products, to conserve resources and minimize dependence on disposal through incineration and landfill.

Recycling of post-consumer products is frequently viewed as the only viable solution for managing municipal waste. However, the very attributes that make plastics a convenient and efficient material can also make it a challenging material to recycle. There are many types of plastic products for packaging applications. Each of the plastics in these products has unique properties which makes it suitable for particular applications, but, all plastics cannot be treated in the same way once disposed.

There has been a significant effort being directed to recycling plastic; since 1990, the number of processing facilities in the post-consumer plastics recycling industry has grown by 81 percent, from 923 facilities in 1990 to 1,677 facilities in 1999 (APC 1999). However, the ratio of recovery to generated plastic waste is still quite low, compared to the ratio of paper and paperboard (51%) and metal materials (57%). "Containers and packaging" comprised the largest portion of waste generated, at 33 percent (76 million tons) of total MSW generation. Only about 10 percent of plastic containers and packaging was recovered in 1999, mostly soft drink, milk, and water bottles (EPA 2001).

Post-consumer recycled (PCR) plastics are already available commercially for packaging of industrial products; however, increased use of PCR for food packaging should be discussed and considered. Lack of knowledge of the substances present in PCR materials has been an obstacle for the application of recycled plastics for food packaging. This is due to the fact that the PCR plastics have a potential to be contaminated toxicologically or with other dangerous substances for human health. Plastics packages are exposed to components from the filled product and after discarding they are exposed to various external influences, as well as to additional thermal stress during the recycling process. As a result, the recycled plastic packaging structures may include compounds that are not intended or suitable for use in food packaging.

PCR plastics intended for use in food packaging may only be used in a manner that complies with the Federal Food, Drug and Cosmetic Act and the Food Additive Regulations (21C.F.R. Part 170) (FDA 1993). In accordance with FDA's "generic" approach to regulating food packaging materials, there are no regulations that deal specifically with the use of recycled material for food packaging. However, the existing regulations can work to ensure the safe use of recycled plastic in food packaging in two ways (PRTF, 1994):

1) Any substance used as a component in a food package must be of a purity suitable for use in contact with food, such that it will not contaminate or adulterate food.
2) Any package component made from recycled materials must comply with the food additive regulations applicable to that component.

There are two primary concerns with the use of recycled plastics, paralleling the two FDA requirements noted above. First, there is a need to ensure that the recycled plastics do not contain substances that could contaminate food. Second, there is the need to ensure that the recycled material does not contain components that have not been cleared by FDA as indirect food additives, or that may be present at levels that are greater than those permitted by FDA regulations.

2.2. Threshold of Regulation

Recycled plastics have been used in food-contact applications since 1990 in various countries around the world. To date, there have been no reported issues concerning health impacts or off-taste resulting from the use of recycled plastics in food-contact applications. This is due to the fact that the criteria that have been established regarding safety are based on extremely high standards that render the finished recycled material equivalent in virtually all aspects to virgin polymers (Bayer 1997).

In the Federal Register (60 FR 36582), the FDA (1995, July 17) established a 'threshold of regulation' process with respect to public health safety concerns. This process was established for determining when the extent of migration to food is so trivial that safety concerns would be negligible. The process exempts substances in food contact materials which result in dietary concentrations at or below 0.5 ppb (μg/kg) from the food additive listing regulation requirement. In other words, dietary exposures to contaminants from recycled food contact plastics packages on the order of 0.5 ppb or less generally are negligible risk. Any substances that may be carcinogens are excluded from this exemption. Based on the extensive study of the toxicological effects of a large number of potentially hazardous chemicals, FDA concluded that the non-carcinogenic toxic effects caused by the majority of unstudied compounds would be unlikely to occur below 1

mg/kg. To provide an adequate safety margin, however, the dietary concentration chosen as a level that presents no regulatory concern should be well below 1 mg/kg. FDA established a dietary concentration of 0.5 μg/kg (0.5 ppb) as the threshold of regulation for substances used in food contact articles. The 0.5 ppb is equivalent to 1.5 micrograms/person/day based on a diet of 1,500 grams of solid food and 1,500 grams of liquid food per person per day (Barlas 1995). A 0.5 μg/kg threshold is 2000 times lower than the dietary concentration at which the vast majority of studied compounds are likely to cause non-carcinogenic toxic effects, and 200 times lower than the chronic exposure level at which potent pesticides induce toxic effects. FDA believes that these safety margins, which are larger than the 100 fold safety factor that is typically used in applying animal experimentation data to humans (21 CFR 170.22), support a conclusion that substances consumed in a dietary concentration at or below 0.5 μg/kg are not of concern (Begley 1997). FDA also concludes that establishing a 0.5 μg/kg dietary concentration as the threshold of regulation is appropriate because it corresponds to a migration level that is above the measurement limit for many of the analytical methods used to quantify migrants from food contact materials. Thus, decisions will usually be made based on dietary concentrations that result from measurable migration into food or food-simulating solvents rather than on worst case estimates of dietary concentration based on the detection limits of the methods used in the analysis.

2.3. Functional Barrier

After several years of deliberations with industry, the FDA published their proposed guidelines for use of recycled plastics, "*Points to consider for the use of recycled plastics in food packaging: chemistry considerations*" (FDA 1992). This document divided plastics recycling into three classes: *Primary* - recycling of plastics which have never had consumer exposure (plant scrap); *Secondary* - the physical cleaning of post consumer plastics by physical processes such as washing and heat treatment (physical process); and *Tertiary* - recycling involving chemical treatment, usually depolymerization of the plastic, purification of the depolymerized moieties, and reproduction of recycled polymer (chemical process).

In general, primary recycling is not considered a safety issue, as contamination of the in-plant scrap is highly unlikely. In this form of recycling, the only consideration is to ensure that materials being used in recycling consist solely of approved food contact materials.

Secondary recycling options have included a variety of approaches to remove contaminants from post-consumer plastic packages or ensure that migration of contaminants to the food product is significantly limited by using "functional barriers".

Tertiary recycling offers a complete cleaning of PCR plastics in the recycling process. For example, the intrinsic structure of the polyethylene

terephthalate (PET) is destroyed by depolymerization, eliminating all potential of any impurities binding to the polymer chains.

One approach to secondary recycling has been to place the post-consumer recycled plastics between two layers of virgin polymer, thus making a multilayer sandwich structure where there is a barrier of virgin polymer between the food and the recycled plastics. The virgin layer in this structure is often called a "functional barrier." Not only do the rate and extent of contaminant migration become very slow, but also the virgin barrier layer that exists between the product and the recycled resin assures consumer safety and product quality (Castle et al. 1996). The use of a functional barrier is a technique which has received "non-objection" status from the FDA in a number of cases (Bakker 1994, Ford 1994).

Even though the barrier layer may not completely eliminate the migration of contaminants from the recycled plastic to the food product, it can sufficiently slow down the process so that essentially there is negligible migration. It is important to determine whether this level is below the threshold of regulation level specified by the FDA.

Most of the functional barriers developed are made of the same polymer as the recycled plastics. Besides the reason that the fabrication process is much easier for the same polymer resin, it is very important to use the same polymers in the structure in order to facilitate further recycling of the package containing PCR materials.

A three or more layered structure in which the recycled polymer is sandwiched between two layers of virgin polymers can be fabricated with a co-extrusion or co-injection process. In 1998, the co-injection blow molded five-layer "Miller" plastic beer bottle was launched commercially. This bottle uses two oxygen barrier layers that employ a specific polyamide formulation, and it is designed to permit incorporation of recycled PET in the middle layer (Powell 1999).

2.4. Recent Developments in Functional Barrier Structure

FDA makes it clear that substances that are separated from food by a functional barrier are not considered to be food additives. Under some circumstances, a layer of virgin polymer can provide a functional barrier, but whether a barrier is functional or not depends on the nature of the polymer, the time and temperature of use, the nature of the food, and the nature and concentration of the contaminants in the recycled core layer.

FDA suggested that some materials may be safely used as a functional barrier, even under the most severe conditions, without any migration from the recycled layer to the food contact layer. Aluminum foil, properly fabricated, is one such material that acts as a functional barrier, preventing migration from the recycled layer into the food product. Hence, a recycled plastic film may be used without any concerns when aluminum foil acts as the food contact layer separating the recycled layer from the food (PRTF, 1994).

However, the same guarantee cannot be given for plastic materials acting as functional barriers because of their relatively inferior barrier properties compared to aluminum.

The FDA has reviewed the use of coextruded laminated structures. Many polyolefins, polyesters, and other polymeric materials of adequate thicknesses have been found to effectively act as functional barriers, preventing significant migration from the recycled layers. Experimental analysis and theoretical modeling have shown that in structures of virgin/recycled PET laminates and structures of virgin/recycled PS laminates, a 1 mil layer of virgin PET and a 1 mil layer of virgin PS, respectively, as the food contact layers, act as functional barriers and allow minimum migration of contaminants into the food when stored for about 2 weeks at room temperature or colder (Thorsheim and Armstrong, 1993).

Continental PET Technologies Inc. (Florence, Kentucky) received a no-objection letter for a co-injected multilayer PET/PCR-PET/PET bottle with a one-mil layer of virgin PET as a functional barrier. It can be used for aqueous, acidic and low-acid foods, including soft drinks and juices that are not hot-filled. Migration tests conducted with this package, using the actual product, proved that the virgin layer was an effective barrier, i.e. it reduced the migration of contaminants into the product to a level that was much below the acceptable daily intake. The results of these tests were submitted to the FDA and the bottle was approved for food contact purposes (Miller, 1993). In 1995, Coca-Cola Co. finally unveiled its famous contour bottle in a three-layer

structure incorporating at least 25% PCR PET between layers of virgin PET (Packworld, 1995)

The no-objection letter validates the principle that a layer of virgin PET can provide a functional barrier, but says nothing about a functional barrier for HDPE. Only recently the FDA has given approval to Union Carbide for use of post-consumer recycled HDPE for packaging dry foods with no surface fats. It must be separated from the food by a food-grade HDPE layer at least 4 mils thick. No letter has been issued for multilayer HDPE bottles, but FDA scientists have suggested that a virgin layer of HDPE might provide a functional barrier for a refrigerated short shelf life product like milk (Bakker 1994).

FP Corp., a Japanese company that wanted to sell trays including post consumer recycled polystyrene, received a "no objection" letter from the FDA indicating that the trays for a New York City restaurant would be okay as long as the recycled layer in the finished trays was separated from the food by a 1-mil thick layer of virgin polystyrene, and the trays would be intended to contact food for no longer than six to eight hours at 50°F or below (Barlas 1995).

2.5. Polyolefins, especially HDPE, as a Functional Barrier

95% of all plastic bottles in the United States market are manufactured from PET or HDPE (48% and 47% respectively). HDPE and PET bottles showed the highest recycling rates of any plastic bottles types, at 23.8 and 22.8 percent respectively. About 29% of recycled HDPE bottles go into making new bottles (APC, 1999). Increased use of PCR HDPE for food packaging should be discussed and considered, not only as a tool for resource use reduction by recycling but also as a marketing tool (Powell 1995).

Franz et al. (1994) presented the results of a study on testing and evaluation of recycled polypropylene with a functional barrier for food packaging. A co-extruded three-layer polypropylene (PP) cup with recycled PP material (R-PP) in the middle layer was investigated with respect to its possible use for a specific food packaging application. The method focused on the direct comparison of the R-PP containing cup with a food grade PP cup manufactured from virgin material. Extraction of the granules indicated a significantly higher migration potential for the recycled materials with recycled vs. new material (R/N) quotients ranging from 6.7 to 8.9. The increased migration potential can not only be recognized from this global view, but also from a specific view based on the numerous additional compounds which can be found in the recycled material in very different concentrations. The extraction results obtained from the finished food contact material (cups)

qualitatively show again the above mentioned differences observed between the granules. However, with measured values of 1.5 - 1.8, the R/N quotients are much lower. With regard to the migration measurements, the differences between R and N are much less pronounced. In the case of global migration, observed R/N quotients are not significantly different from 1.0, when the precision of the analytical method is taken into account. Based on the test results, Franz et al. concluded that the virgin PP contact layer was able to accomplish its function as a barrier layer against the recycled PP material in the middle layer, under the intended conditions of use.

From the recent alternative approach for calculating the worst case diffusion coefficients (Baner at al. 1994), it can be estimated that the relative values of diffusion coefficients in the polyolefins follows the sequence: $D_{LDPE}=20 \cdot D_{HDPE}$ and $D_{LDPE}=150 \cdot D_{PP}$, and the migration rate for HDPE is a factor of 2 - 3 higher than PP ($\sqrt{D_{HDPE}} = \sqrt{7.5 \cdot D_{PP}}$). HDPE shows a slightly higher diffusion coefficient than PP; however, it can be assumed, from the results of the above study, that HDPE also has an ability to be an effective functional barrier. Polyolefins such as HDPE have less barrier ability than PP and therefore the deficiencies of the Begley and Hollifield model are a serious problem.

2.6. Evaluation of Migration through a Functional Barrier

2.6.1. Migration Studies

Using the threshold of regulation concept, the FDA has pioneered an approach to assess the safety of unknown contaminants in recycled plastic feedstreams. This approach, although ultra-conservative, has provided a means for industry and regulators to work together to achieve the goal of reducing municipal solid waste and thereby protecting our environment. And FDA decisions will usually be made based on dietary concentrations that result from measurable migration into food or food-simulating solvents rather than on worst-case estimates of dietary concentration based on the detection limits of the methods used in the analysis. Eventually, using the threshold policy to evaluate food package safety requires the amount of migration to be determined. And more importantly, the FDA has accepted the use of model simulants and an analytical model (see Equation 2.9) as the basis for approvals of the use of functional barriers.

Traditionally, migration tests are performed by using food simulants such as water, edible oils, aqueous ethanol solutions or sometimes food. These tests are time consuming in two ways: generally the accelerated tests run for at least 10 days, and detection of the migrants at low concentrations in the food is generally difficult. These analyses are also expensive and may generate hazardous laboratory waste, depending on the analytical methods. A number of studies, some of them highly time-consuming, have been

published for determining the transfer of a contaminant into the food and thus for evaluating the efficacy of the functional barrier (Jabarlin and Kollen 1988, Seyler 1990, Boven et al. 1991, Khinava and Aminabhavi 1992, Miltz et al. 1992, Franz et al. 1994, Piringer et al. 1998).

Other studies of interest were of a theoretical basis, and considered the transport of the contaminant through the polymer by Fickian diffusion (Ilter et al. 1991, Begley and Hollifield 1993, Laoubi et al. 1995, Laoubi and Vergnaud 1996, Katan 1996, Piringer et al. 1998).

In the 1970s, Figge (1972, 1980) studied the migration of antioxidants from HDPE, PVC and PS into food oils and fat simulants. These studies showed that migration to the oils and fat simulants was predictable. The migration of the antioxidant butylated hydroxytoluene (BHT) and two other hydrocarbons from polyolefins into food and food simulants was studied (NBS 1982). These studies also concluded that migration of antioxidants is predictable. In another extensive study (A. D. Little 1983) the migration of antioxidants, styrene monomer and plasticizer were measured from various polymers such as PE, PS, PVC and EVA into many food simulating liquids and foods. From this study it can be concluded that migration is predictable. A detailed migration study by Gandek et al. (1989) showed migration could be accurately predicted if all the migration controlling parameters are known. This study used measured values for the diffusion coefficient of the additive in the polymer, the mass transfer coefficient (agitating coefficient), partition coefficient of the additive between the polymer and the food, and the reaction

rate constant for the degradation of the additive in the food (which affects the partition coefficient). In a general sense, all of the detailed migration studies mentioned above have shown that migration is controlled by diffusion through the polymer according to Fick's 2nd Law, and diffusion follows Arrhenius behavior with temperature.

Generally, the coefficient of mass transfer at the polymer-liquid interface was assumed to be infinite, leading to simple but particular equations. An interesting study on the effect of surface barriers was made by considering the diffusion of a substance from a polymer into a bath system of finite volume (Etters 1991). It must be pointed out that a great number of studies have been reported on the subject of mass transport between a polymer and a liquid.

In order to correlate all these results, it is necessary to examine the process in depth and to obtain a general solution to the problem. The process of contaminant transfer is rather complex, and appropriate assumptions have to be made in order to clarify the problem.

2.6.2. Modeling for a Single Layer Structure

Mathematical models are often used to describe or to predict migration of monomers and additives from plastic packaging materials to food (Figge 1980, Reid et al. 1980, Chatwin and Katan 1989, Piringer 1994, Vergnaud 1995/96). A very large number of parameters can influence migration: the structure of the migrant, its affinity to the food or food simulant, as well as the

interaction of the food or food simulant with the polymer. Due to this complexity, simplified equations are often used.

The models used for prediction of migration through an elastic polymer, above its glass transition temperature, are based on the diffusion equation, described as Fick's second law, based on one-directional diffusion:

$$\frac{\partial C_{x,t}}{\partial t} = D\frac{\partial^2 C_{x,t}}{\partial x^2} \qquad (2.1)$$

Different boundary conditions used for integration of Equation 2.1 have been proposed and classified (Crank 1975, Vergnaud 1991). The models differ according to the assumptions made about the concentration of the migrant in the food and in the package. A model assuming that the concentration of the migrant can be considered as constant leads to very simple equations; it is usually called "infinite packaging." Likewise a model assuming that the volume of food is so large that the concentration of the migrant is regarded as constant, often 0, is called "infinite food."

Analytical models of packaging with finite thickness, which is the closest to reality, involve a packaging material of finite thickness, in contact on one side with food or food simulant (Miltz 1992, Vergnaud 1995/96). Two subclasses can be distinguished, depending on the volume of food: 1) finite packaging, finite food, 2) finite packaging, infinite food.

With one-directional transfer, with finite volumes of food and of package when the coefficient of convective transfer in the liquid is infinite (well agitated), the solution is given by Equation 2.2.

$$\frac{M_{F,t}}{M_{F,e}} = 1 - \sum_{n=0}^{\infty} \frac{2\alpha(1+\alpha)}{1+\alpha+\alpha^2 q_n^2} \exp\left[\frac{-Dq_n^2 t}{L^2}\right] \tag{2.2}$$

where the q_ns are the non-zero positive roots of $tan\ q_n = -\alpha q_n$.

Assuming that the volume of food is infinite for finite packaging, the solution of Equation 2.1 can then be expressed either by Equation 2.3 or by Equation 2.4 (Miltz 1992, Vergnaud 1991):

$$\frac{M_{F,t}}{M_{F,e}} = 1 - \sum_{n=0}^{\infty} \frac{8}{(2n+1)^2 \pi^2} \exp\left[-\frac{(2n+1)^2 \pi^2}{4L^2} Dt\right] \tag{2.3}$$

$$\frac{M_{F,t}}{M_{F,e}} = 2\left(\frac{Dt}{L^2}\right)^{0.5} \left\{\frac{1}{\pi^{0.5}} + 2\sum_{n=1}^{\infty} (-1)^n ierfc\left[-\frac{nL}{(Dt)^{0.5}}\right]\right\} \tag{2.4}$$

The hypothesis of an infinite food is equivalent to claiming that the concentration of the migrant is constantly equal to zero in the food. When the volume of the food is much larger than that of the polymer, Equations 2.2, 2.3, and 2.4 should give equivalent results. For long contact times ($M_{F,t}/M_{F,e} > 0.5$ - 0.6), Equation 2.3 can be reduced to Equation 2.5 (Hamdani et al. 1997):

$$\frac{M_{F,t}}{M_{F,e}} = 1 - \sum_{n=0}^{\infty} \frac{8}{\pi^2} \exp\left[-\frac{\pi^2}{L^2} Dt\right] \tag{2.5}$$

For short contact times ($M_{F,t}/M_{F,e} < 0.5$-0.6), Equation 2.4 can be reduced to Equation 2.6, which is widely used (Hamdani et al. 1997):

$$\frac{M_{F,t}}{M_{F,e}} = \frac{2}{L}\left(\frac{Dt}{\pi}\right)^{0.5} \tag{2.6}$$

Several models of packaging with infinite thickness for different situations of materials in contact with food were developed (Reid et al. 1980). Some equations correspond to the case of a material with infinite thickness in contact with a well-agitated food or food simulant. Of course, the concept of infinite thickness is an oversimplifying assumption.

If the food is also considered as infinite for infinite packaging, the solution of Equation 2.1 is given by Equation 2.7 :

$$\frac{M_{F,t}}{A} = 2C_{P,0}\left(\frac{Dt}{\pi}\right)^{0.5} \qquad (2.7)$$

If there is a limited volume of food or food simulant, the solution of the differential Equation 2.1 is Equation 2.8:

$$M_{F,t} = C_{P,0}AK(1 - e^{z^2}erfcZ) \qquad (2.8)$$

where $Z = \frac{(Dt)^{0.5}}{AK}$

In cases where Z is small (Z < 0.05), for short exposure times, Equation 2.8 reduces to Equation 2.7, which is very practical since $C_{p,0}$ can be determined quickly, by an extraction method. It has even led to a predictive approach of worst case migration, based on worst case diffusion coefficients (Baner et al. 1996). Since the packaging material is considered as infinite, the model can only overestimate migration, which gives the biggest safety margin for consumers' protection. However, if the safety margin is too great, those equations are not very useful.

2.6.3. Modeling for a Multi Layer Structure with Functional Barrier

2.6.3.1. Analytical Approach

Begley & Hollifield (1993) treated the infinite packaging and infinite food conditions mathematically as a pseudo-membrane problem where there is a fixed concentration on one side of the film and zero concentration on the other side. The solution to this problem is expressed in Equation 2.9 (Crank 1975), which assumes that $C_{p,o}$ remains constant with time, and the food is an instantaneous and infinite sink for the contaminant. The FDA has accepted the use of model simulants and the Begley and Hollifield diffusion model (Equation 2.9) as the basis for approvals of the use of functional barriers.

$$\frac{M_{F,t}}{M_{F,e}} = \frac{Dt}{(L-R)^2} - \frac{1}{6} - \frac{2}{\pi^2}\sum_{1}^{\infty}\frac{(-1)^n}{n^2}\exp\left(\frac{n^2\pi^2 Dt}{(L-R)^2}\right) \qquad (2.9)$$

In reality, Equation 2.9 is an oversimplification of the diffusion process for a mutilayer package because: (1) the contaminant cannot freely move into the virgin layer, it must diffuse out of the recycled layer; and (2) concentration of the contaminant will decrease with time because of diffusion into the virgin layer or out of the opposite side of the package.

The problem of mass transfer from a package of finite thickness into a liquid can be resolved by a mathematical treatment only in two cases (Crank 1975, Vergnaud 1991): (1) when the volume of liquid is finite and the coefficient of mass transfer at the interface is infinite, and (2) when the coefficient of mass transfer at the interface is finite, and the volume of liquid is

infinite, which is unrealistic. For the former case, Laoubi and Vergnaud (1996) presented the following equation:

$$\frac{M_{F,t}}{M_{F,e}} = 1 - \frac{8}{\pi^2}\frac{L}{R}\sum_{1}^{\infty}\frac{(-1)^n}{(n+1)^2}\sin\left[\frac{(2n+1)\pi R}{2L}\right]\exp\left[-\frac{(2n+1)^2\pi^2 Dt}{4L^4}\right] \quad (2.10)$$

Etters (1991) attempted to mix the analytical expressions obtained with these two cases in a rather simple way.

Katan (1996) suggested a completely different approach to this problem. His solution to the prediction of the migration through a functional barrier structure was based on the assumption that the virgin layer acts like a "sponge." According to this theory, at the burst-out time, a substantial portion of the contaminant diffuses into the barrier, and then migrates into the food at an accelerating rate. It states that in this situation, the thickness of the barrier layer has more effect on the burst out time than the diffusion coefficient of the compound. That is, the diffusion coefficient does not play as much of a role as the thickness of the polymer in reducing the migration. The mathematical treatment gives the concentration of the contaminants in the food contact layer at t_∞ when concentrations are equal in the core and barrier layer but before migration into the food as:

$$C_\infty = \frac{C_0 x_R}{x_R + x_B} \quad (2.11)$$

where x_R and x_B, are the thickness of the core and barrier layers respectively. C_0 is the initial concentration of the contaminant, and C_∞ is the concentration of the contaminant in the food contact barrier layer at

equilibrium.

Franz et. al. (1997) presented another approach in multilayer migration modeling. A theoretical migration model (Equation 2.12) was developed and experimentally verified in their work using artificially contaminated multilayer HIPS sheets with different functional barrier thicknesses.

$$\frac{M_{F,t}}{A} = \frac{2}{\sqrt{\pi}}\left[C_{P,e}\left(1+\frac{L-R}{R}\right) - C_{B/2}\frac{L-R}{R}\right]\rho\sqrt{D}\left(\sqrt{\theta_r + t} - \sqrt{\theta_r}\right) \quad (2.12)$$

θ_r, the lag time, is calculated to take into account diffusion during co-extrusion.

It was concluded that HIPS plastic was appropriate under given application parameters for functional barrier packaging design, and a functional barrier thickness range of 100 to 200 μm was found to be a very efficient one for HIPS, even in the case of the exaggerated test conditions applied. One other important conclusion was also drawn. The definition of the functional barrier efficiency by means of the lag time only, as Katan did, turns out to be inappropriate, because even after having the breakthrough of contaminants to the food contact surface, the remaining functional barrier efficiency can be high enough to prevent migration of amounts into the foodstuff which raise toxicological concerns.

2.6.3.2. Numerical Treatment

The process of mass transport from a package into a food, especially through the functional barrier, is complex. Often a mathematical treatment is not feasible and numerical methods can be used for resolving the problems.

Laoubi and Vergnaud (1996, 1997) presented a solution of this situation by using a numerical model with finite differences, taking all the known facts into account. The planar package is divided into N parallel slices of thickness Δx and each slice is associated with the integer n. The new concentration in the package after elapse of time Δt, C_N, is expressed in terms of the previous concentrations at the same and adjacent places. The amount of contaminant in the food at any time can be obtained by:

$$M_{F,t} = N \cdot \Delta x \cdot C_{in} - \Delta x \left[\sum_{n=1}^{N-1} C_{n,t} + \frac{1}{2}(C_{0,t} + C_{N,t}) \right] \qquad (2.13)$$

where, C_{in} is initial concentration of the contaminant in the polymer.

Numerical procedures such as the finite element and finite difference methods can be excellent alternatives to predict solutions to migration problems, whenever the analytical approach is difficult to apply to a particular problem regarding the simulation of the mass transfer process. The finite element method is of particular importance, due to its wider applicability as compared to finite differences. The finite difference method is rather cumbersome when regions have curved or irregular boundaries, and it is difficult to write general computer programs for the method (Segerlind 1984).

It is impossible to document the exact origin of the finite element method because the basic concepts have evolved over a period of more than 150 years. Although the origin of the method is vague, its advantages are clear. The method is easily applied to irregular-shaped objects composed of several different materials and having mixed boundary conditions. These methods have been widely used for more than decades by engineers interested in stress problems and in steady-state heat flow. More recently, mathematicians have attempted to put the methods on a rigorous mathematical basis, including their use for time dependent unsteady-state problems.

The finite element method has been applied to heat conduction and fluid dynamics problems since it was proposed (Lewis et al 1981, 1996, Reddy 1994). However, little work has been done regarding its application to mass transfer or diffusion problems in food packaging plastics or migration of substances to the food. There is general awareness among scientists and engineers that the phenomena of heat flow and diffusion are basically the same, so that methods applied to heat transfer problems can be incorporated into migration problems with the changes of notation needed to transcribe from one set of solutions to the other (Crank 1975).

3. MATERIALS AND METHODS

3.1. Materials

3.1.1. Film Samples

Experiments were carried out using single and multilayer high density polyethylene (HDPE) films provided by The Dow Chemical Co. (Freeport, Texas, USA). Three layer co-extruded HDPE films, which had a symmetrical structure with the core layer contaminated by known amounts of contaminant simulant and virgin outer layers as the functional barriers, were fabricated with varying thickness of the outer layers.

For the single layer film and the core layer of the three layer structures, low density polyethylene (LDPE) masterbatch resin (DOWLEX 92.48, density=0.917 g/cm^3, MI=2.3), which had 10% concentration of contaminant simulants (100 mg/g), was diluted with HDPE (DOW 05862N, density=0.9625 g/cm^3, MI=5) by dry blending to have 1% (10 mg/g) concentration of contaminant simulants, so the single layer and the core layer of three layer structures consisted of 10% (w/w) of LDPE and 90 % (w/w) of HDPE. The same HDPE, DOW 05682N, was used to coextrude the outer layers of the 3-layer structures and to make a single layer film for partition experiments.

The thickness of each layer was controlled precisely by automatic devices. During the co-extrusion, the various running parameters were adjusted such that outer layers with differing thicknesses of 0.6 mil (0.0015

cm), 1.2 mil (0.003 cm) and 1.8 mil (0.0045 cm) were prepared, and the thickness of the core layer was kept at 1.2 mil (0.003 cm). Total thickness of the single/ multilayer polymer samples was measured by MAGNA-MIKE® Model 8000 Hall Effect Thickness Gage (Panametrics, Waltham, MA) and Micrometer Model 549 M (Testing Machines, Inc., Amityville, NY). The details of film samples will be discussed in the "Results and Discussion" section.

Film samples were received as roll stocks (200' x 22" for each roll). The rolls were stored at a temperature below – 25°C during the experiment to avoid/reduce undesirable mass transfer between the core and outer layers.

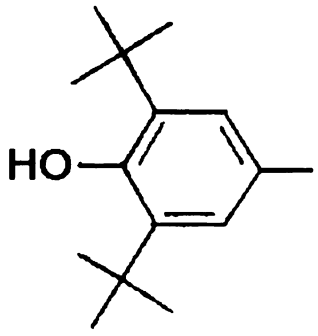
3.1.2. Contaminant Simulants

As the contaminant simulants, three phenolic antioxidants: BHT (3,5-di-*t*-butyl-4-hydroxytoluene) (Eastman Chemical Co. Kingsport, TN), Irganox™1076 (octadecyl-3-(3,5-di-*t*-butyl-4-hydroxyphenyl)-propionate), and Irganox™1010 (pentaerythritol tetrakis (3-(3,5-di-*t*-butyl-4-hydroxyphenyl) propionate)) were selected initially since they are easy to detect and analyze (Irganox™ is a brand name of Ciba Specialty Chemicals Co., Tarrytown, NY). However, Irganox™1010 was excluded later due to its extremely slow mass transfer characteristics. Experimental results of the BHT and Irganox™1076 migration tests will be presented and discussed. Figure 3.1 shows the chemical structures of these three compounds, and selected properties are listed in Table 3.1.

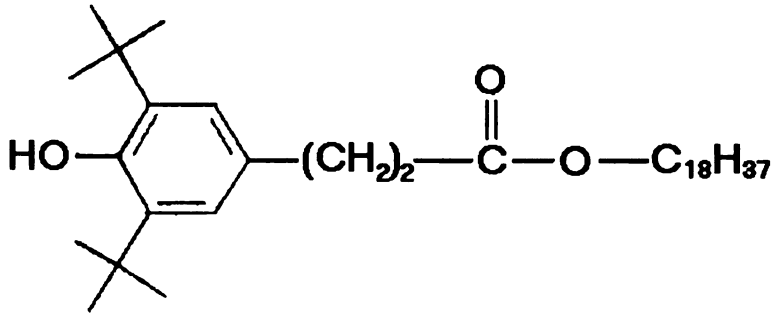
3.1.3. Food Simulants

For this study, two food simulants were selected: absolute (100%) ethanol and 50% aqueous ethanol. These simulants are recommended by FDA as the appropriate alternative food simulants for replacing oil and fat and low/high alcoholic foods, because they are easy to handle and analyze (FDA, 1995).

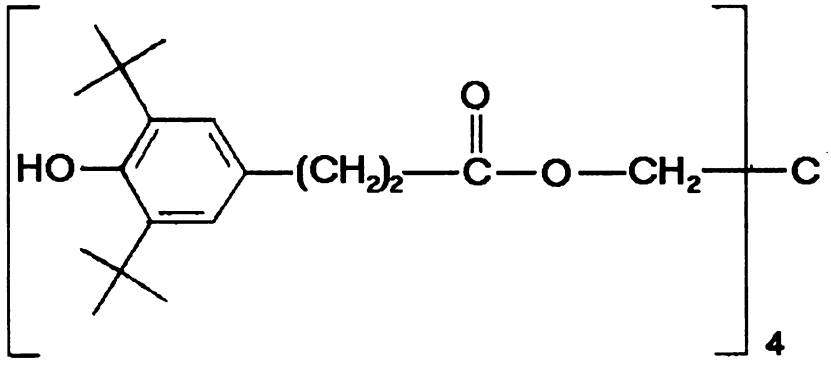
The absolute ethanol (ethyl alcohol, HPLC grade, density: 0.785 g/cm^3, Sigma-Aldrich, Milwaukee, WI) was diluted with water (HPLC reagent grade, J.T. Baker, Phillipsburg, NJ) to make the 50% aqueous solution.



BHT



Irganox™1076



Irganox™1010

Figure 3.1 Chemical structures of antioxidants used as contaminant simulants

Table 3.1 Properties of antioxidants used as contaminant simulants

Simulant	BHT	Irganox™1076	Irganox™1010
Molecular weight	220	531	1178
Molecular formula	$C_{15}H_{24}O$	$C_{35}H_{62}O_3$	$C_{73}H_{108}O_{12}$
Melting point, °C	70	50-55	110-125
Boiling point, °C	265		
Density (at 20°C)	1.048	1.02	1.15
Flash point, °C	127	273	297
Solubility (at 20°C)			
Benzene	40	57	
Acetone		19	47
Chloroform		57	71
Ethanol	30	1.5	1.5
Toluene		50	60
Methanol	20	0.6	0.9
Water	Few ppm	< 0.01	< 0.01

3.2. Methods

3.2.1. Chromatographic Analysis

Concentrations of the phenolic antioxidants in the film samples and food simulants were analyzed by reverse phase high performance liquid chromatography (HPLC). The HPLC system consisted of a Waters Model 150-C ALC/GPC and a Waters 486 Tunable Absorbance Detector which was interfaced to a Waters 730 Data Module for quantification (Waters Corporation, Milford, MA). The chromatographic conditions were as follows:

Column: Delta Pak™, C_{18}, 300 A^{o}, 3.9 x 150 mm Column, Part No. 35571 (Waters Corporation, Milford, MA)

Solvent Systems:

BHT – 85% Methanol (HPLC solvent grade, J.T. Baker)/15% Water

Irganox™1076 – 100% Methanol

Irganox™1010 – 97.5% Methanol/2.5% Water

Flow Rate: 1.0 ml/min

Detector Wavelength: 280 nm

Injection: 20 μl volume by automatic sample injection system which can be loaded with sixteen 4 ml glass vials with screw top (Supelco, Bellefonte, PA) and PTFE septums (Waters, Milford, MA)

Peak areas and retention times were determined by the computing integrator with the following conditions: Pick Width: 20, Noise Reduction: 10

and Area Rejection: 100. Quantitative analysis was made by the external standard calibration method, i.e. the retention time and area responses were compared with known concentrations of standards. The standard calibration curves for determining the concentrations of BHT and Irganox™1076 are included in Appendix A.

3.2.2. Migration Cell and Experiments

Migration experiments were performed in accordance with ASTM D 4754 – 93, "Two-Sided Liquid Extraction of Plastic Materials Using FDA Migration Cell" (ASTM 1995) with some modifications in the migration cell. A schematic of the migration cell is presented in Figure 3.2. It was prepared as follows: 20 plastic test specimens in the form of round disks, 2.15 cm (21.5 mm) diameter for each disk (with a total surface area of 145.22 cm^2; edges neglected), were punched out from the plastic sample. Then the test specimens were weighed and threaded onto a stainless steel wire with alternating glass tube cutting spacers (4mm diameter and 2 mm height; cut from glass tube) to prevent the specimens from contacting each other. The threaded specimens on the wire were placed in a 40 ml amber glass vial with screw top (29 mm diameter and 81 mm height, Supelco, PA). The food simulant was added to the vial to its shoulder, a volume of 36 ml, and the vial was capped with a Polypropylene Hole Cap with PTFE/ Silicone Septum (Supelco, PA).

Assembled migration cells were stored in a controlled atmosphere room maintained at 23°C, as well as in ovens maintained at 31°C and 40°C. Aliquots of the food simulant were removed from the migration cell at various time intervals (initially about 30 minutes after the start of the experiment and continued to a total time of several months), which were determined by preliminary examinations, and the concentration of the contaminant simulants in the food simulant were determined by the HPLC method as described above. No attempt was made to shake or agitate the migration cell during storage.

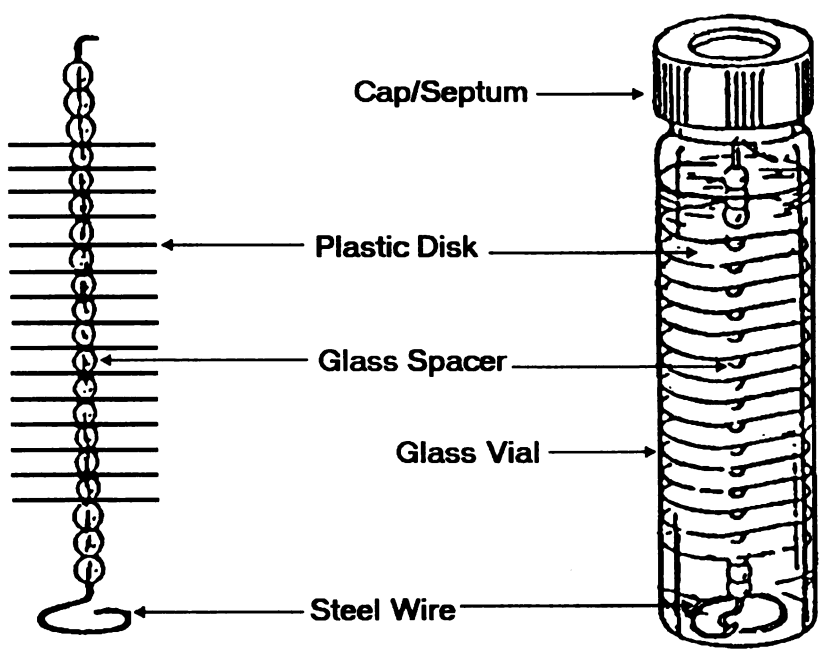


Figure 3.2 Migration cell

3.2.3. Determination of Initial Concentration of Contaminant Simulant

To determine the initial concentration of contaminant simulant in the plastic sample, the Soxhlet extraction method was applied for the plastic samples with thickness of 2.4 mil (0.006 cm) and lower; however, for the thicker samples with thickness of 3.6 mil (0.009 cm) and higher, the migration cell extraction method was applied, since the recovery rate from Soxhlet extraction was not high enough to determine the initial concentration due to the thickness of the samples.

For Soxhlet extraction, 2 g of film samples were cut into small pieces and extracted with 150 ml of acetonitrile (HPLC grade, EM Science, Gibbstown, NJ) for BHT and methylene chloride (HPLC solvent, J.T. Baker) for Irganox™1076 in a Soxhlet extraction apparatus for 14 hours. The extracts then were filtered through a 0.45 μm porosity HV filter (Japan Millipore, Yonezawa, Japan) and analyzed by HPLC.

The identical migration cell and experimental methods, explained above, were used to determine the initial concentrations of the contaminant simulants in thicker film samples. When the extraction process reached equilibrium, samples of the food simulants were taken and injected in the HPLC for quantification and the remaining food simulants were discarded. The plastic disks were then rinsed with ethanol and blotted dry with a lab tissue, and were again analyzed by the Soxhlet extraction method for complete extraction.

3.2.4. Determination of Diffusion Coefficient

The migration cell extraction method and HPLC as an analytical method were used to determine the diffusion coefficients of the contaminant simulants in the plastics.

In general, at the start of any migration experiment, the polymer can be treated as if it were infinitely thick. Therefore, simpler mathematical expressions can be used to model this "early-time" diffusion-controlled behavior where less than 50 – 60 % of the contaminant simulant has migrated. With the assumptions of constant diffusion coefficient and no mass transfer resistance between the plastic and food simulant, the mathematical equation can be expressed as Equation 2.7 (Crank 1975, Reid et al. 1980):

$$\frac{M_{F,t}}{A} = 2C_{P,0}\left(\frac{Dt}{\pi}\right)^{0.5} \qquad (2.7)$$

where,

A : *Surface area of polymer in contact with food, cm^2*

$C_{P,0}$: *Concentration of the contaminant within the polymer at time 0 (before contact) – Initial concentration, g/cm^3*

D : *Diffusion coefficient of the contaminant within the polymer, cm^2/sec*

$M_{F,t}$: *Mass that has migrated into the food after time t for unit area of polymer, g*

t: *Time, sec*

The plot of extraction data up to 50 – 60 % migration as a function of the square root of time showed a straight line, and the slope of the line was used to determine the diffusion coefficients.

Assembled migration cells were stored in a controlled atmosphere room maintained at 23°C, as well as in ovens maintained at 31°C and 40°C. Aliquots of the food simulant were removed from the migration cell at various time intervals, initially about 30 minutes after the start of the experiment and continued to reach equilibrium (complete migration). The concentrations of the contaminant simulants in the food simulants were determined by the HPLC methods as described above. No attempt was made to shake or agitate the migration cell during storage.

3.2.5. Determination of Partition Coefficient

The migration cell extraction method and HPLC as an analytical method were also used to determine the partition coefficients for the HDPE/ contaminant simulant/food simulant systems used.

The HDPE (DOW 05862N) single layer film sample with a thickness of 1.17 mil (0.003 cm) was provided by The Dow Chemical Co. The initial concentration of the contaminant simulant in the polymer sample was zero.

The same extraction cell samples were assembled as explained in the "Migration Cell and Experiments" section and the food simulants containing 100 ppm (g/cc) of contaminant simulant were added. The assembled cells were stored in a controlled atmosphere room maintained at 23°C, as well as

in ovens maintained at 31°C and 40°C. Two replicates of blanks which had only the food simulants, with the same concentration of the contaminant simulant, in vials containing the support stand with glass tube cutting spacers were stored at the same conditions.

After mass transfer (sorption by the polymer) reached equilibrium, both the food simulants and the plastics were analyzed for contaminant simulant content, and partition coefficients were calculated for each sample.

The amount of contaminant simulant in the polymer phase was calculated by subtracting from the amount of contaminant simulant in the blank food simulant solution (at initial time) the amount of contaminant simulant in the food simulants after equilibrium (at final time). The contaminant simulant concentration in the polymer at equlibrium was calculated according to Equation 3.1 (Gavara et al, 1996):

$$C_{P,e} = \frac{(C_{F,0} - C_{F,e})m_F}{m_P} \tag{3.1}$$

where,

$C_{P,e}$: *Concentration of the contaminant simulant in the polymer at equilibrium, g/g*

$C_{F,0}$: *Concentration of the contaminant simulant in the food simulant at time 0 (Initial), g/g*

$C_{F,e}$: *Concentration of the contaminant simulant in the food simulant at equilibrium (final), g/g*

m_F: *Mass of the food simulant in the migration cell, g*

m_P: *Mass of the polymer sample in the migration cell, g*

Once the equilibrium concentrations of the contaminant simulant in both the polymer and food simulants were obtained, the partition coefficient, *K*, was calculated according to Equation 3.2:

$$K = \frac{C_{P,e}}{C_{F,e}} \quad \left[\frac{g/g}{g/g}\right] \tag{3.2}$$

4. MODELING AND COMPUTER PROGRAMMING

4.1. Modeling Concepts

The first purpose of the study was to develop an adequate model which describes in detail the process of transport of a substance through a package into a liquid food when the package is made of two polymer layers: a recycled core layer in good contact with a virgin outer layer. Initially, the concentration of the diffusing substance, referred to as the contaminant, is uniform in the recycled layer, while the virgin polymer is free from contaminant.

Various problems of diffusion in a multilayer structure comprised of two layers for which the diffusion coefficients are different have been solved (Crank 1975). Carslaw and Jaeger (1959) solved problems on conduction of heat in composite solids for several cases, which provide solutions to diffusion problems with appropriate substitution of variables. The solutions are similar in form to those presented in the "Literature Review" chapter but obviously more complicated. The solutions become further complicated by practical problems such as diffusion in 3 or more layers, migration with partitioning, diffusion during extrusion and storage of polymers, and migration with back-diffusion by food. In such cases, analytical solutions become very difficult to use. Therefore a numerical method, the finite element method, was selected in this study to permit a simpler and more efficient approach to solving these complicated problems.

4.1.1. Assumptions for Modeling

The basic concept is diffusion of the contaminant symmetrically from the core layer through the barrier layers and into the food, with no resistance to transfer at the virgin layer/food interface (infinite mass transfer coefficient at the interface). Therefore migration can be modeled as one-dimensional and the polymer represented by half the structure: from the center of the core layer to the outside of the virgin layer. Additional assumptions include:

1) The two packaging layers are in perfect contact. The coefficient of mass transfer through the interface between the two polymers is infinite, as the two films are in perfect contact.
2) A single contaminant is migrating from the polymer and it is uniformly distributed throughout the core layer initially.
3) The diffusion coefficient of the contaminant in a specific polymer depends only on temperature, as is often the case when the concentration of the diffusing substance is low (Crank 1975, Vergnaud 1992).
4) One-dimensional mass transfer in the polymer is considered since the thickness of the polymer is small in comparison to the surface area of the polymer.
5) The food is considered well-mixed and of infinite volume for calculating migration without partitioning; this is reasonable as the ratio of volume of food to volume of polymer is large enough, and it is usually achieved in packaging practice.

6) The concentration of contaminant in the core layer declines as migration proceeds; finite packaging.

7) Partitioning of the contaminant between the polymer and food can be modeled using a time-independent partition coefficient.

8) There is no interaction between the polymer and the food that affects diffusion of the contaminant, e.g. alcohols do not cause swelling of polyolefin (Becker et al. 1983).

The contaminant diffuses through the polymer and reaches the polymer/food interface, where it will be transferred immediately into the food with a constant zero concentration of the contaminant at the polymer/food interface.

Therefore the initial and boundary conditions are (refer to Figure 4.1):

$t = 0, \quad 0 < x < R, \quad C = C_i$

$R < x < L, \quad C = 0 \quad$ and

$t \geq 0, \quad$ at $x = 0, \quad \frac{\partial C}{\partial x} = 0$

at $x = R, \quad C_{PCR} = C_{FB}$

at $x = L, \quad C_{FB} = 0$

where,

C_i : *Initial concentration of the contaminant in the core layer*

C_{PCR} : *Concentration of the contaminant in the core layer at time t*

C_{FB} : *Concentration of the contaminant in the outer layer at time t*

The transfer rate and amount are controlled not only by the diffusion coefficient but also by the partition factor. To incorporate the partition coefficient in the computational model, it is converted to percent partition which provides a measurement of the maximum extent of migration in a percent value for a specific system at any time. Migration with partitioning at any time t is then calculated as migration without partitioning multiplied by percent partition.

4.1.2. Parameters for Modeling

The scheme of the system is presented in Figure 4.1 and the following parameters are considered for modeling:

1) the initial concentration of contaminant in the core layer,
2) the thicknesses and surface areas, and densities of the two layers of the packaging,
3) the volume and density of the food simulants,
4) the diffusion coefficients of contaminants in the polymer layers; core and outer layers, and
5) the partition coefficient of the contaminant between the outer layer and the food simulants.

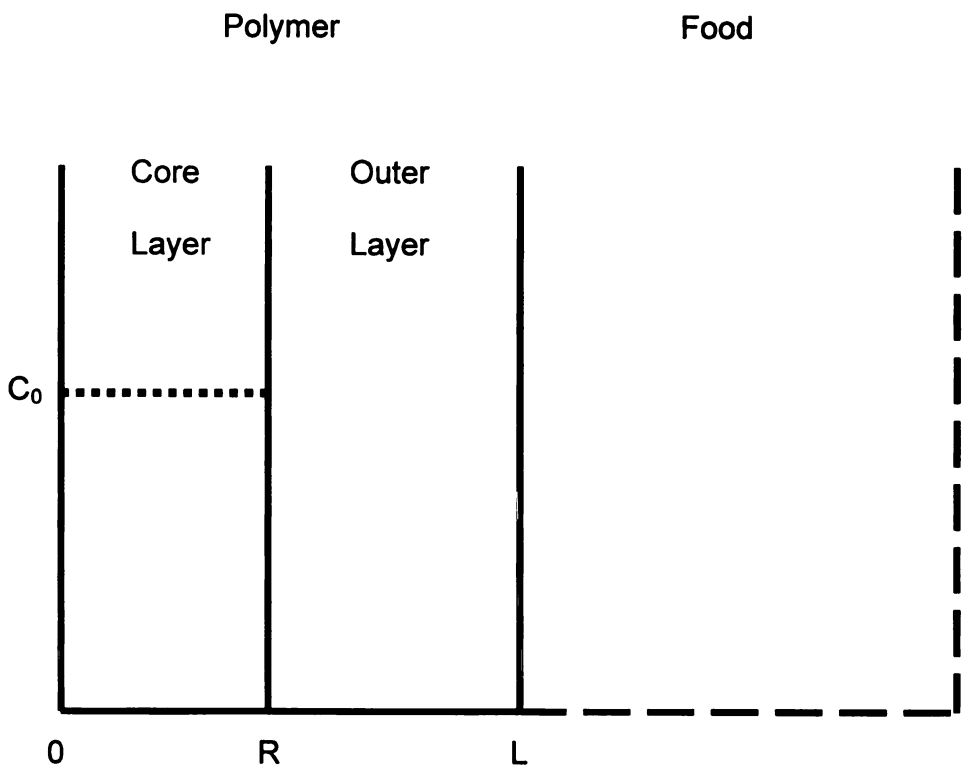


Figure 4.1 Scheme of system for modeling

4.2. Modeling by the Finite Element Method

The finite element method is a numerical procedure for solving physical problems governed by a differential equation. It can be described as an adaptation of the calculus of variations, and results in a system of simultaneous linear or nonlinear equations. The number of equations is usually very large, so the method has practical value only with use of computer technology. The finite element method has two characteristics that distinguish it from other numerical procedures (Segerlind, 1984): 1) The method utilizes an integral formulation to generate a system of algebraic equations; 2) The method uses continuous piecewise smooth functions for approximating the unknown quantity or quantities.

The following description of the Finite Element Method (FEM) was derived from the work of Segerlind (1984). The governing differential equation for one-dimensional unsteady-state diffusion through the polymer is expressed by Fick's 2nd law (Equation 2.1) as follows:

$$\frac{\partial C_{x,t}}{\partial t} = D\frac{\partial^2 C_{x,t}}{\partial x^2}$$

where,

D : *Diffusion coefficient*

$C_{x,t}$: *Concentration of the contaminant within the polymer at position x and time t*

x : *Position or thickness term*

t : *Time*

The concentration of contaminant, C, has a single value for a particular value of x (position or thickness term). The boundary conditions associated with Equation 2.1 are usually specified concentrations. This equation has the following general form used in applying the finite element method where $Q = 0$ and $\lambda = 1$:

$$D\frac{\partial^2 C}{\partial x^2} + Q - \lambda\frac{\partial C}{\partial t} = 0 \qquad (4.1)$$

where Q and λ are the constants.

In the finite element method, the general procedure for analyzing Equation 4.1 is to evaluate the Galerkin residual integral with respect to the space coordinates for a fixed instant of time. This yields a system of differential equations that are solved to obtain the variation of C with time.

In the following equations, all of the matrices and vectors are enclosed in brackets. Column vectors are designated by curly brackets ({ }), row vectors and matrices are designated by square brackets ([ ]), and the superscript (e), $^{(e)}$, in the brackets represents an "element".

The residual equation is

$$\{R^{(e)}\} = -\int_{X_i}^{X_j} [W]^T \left(D\frac{\partial^2 C}{\partial x^2} + Q - \lambda\frac{\partial C}{\partial t}\right)dx = 0 \qquad (4.2)$$

where $[W]^T$ contains the Galerkin weighting functions.

Defining $[W]^T = [N]^T$, the vector containing the element shape functions, and applying the solution method of the lumped formulation, Equation 4.2 can be written as

$$\{R^{(e)}\} = [k^{(e)}]\{C^{(e)}\} - \{f^{(e)}\} + [p^{(e)}]\{\overline{C}^{(e)}\} = \{0\} \tag{4.3}$$

where $\{R^{(e)}\}$ is the contribution of element *(e)* to the final system of equations.

This contribution consists of an element stiffness matrix $[k^{(e)}]$ and an element force vector $\{f^{(e)}\}$. The element matrices are the important results and they are

$$[k^{(e)}] = \frac{D}{L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \qquad \text{and} \tag{4.4}$$

$$\{f^{(e)}\} = \frac{QL}{2}\begin{Bmatrix} 1 \\ 1 \end{Bmatrix} \tag{4.5}$$

The element force vector $\{f^{(e)}\}$ will be zero since the governing equation (Equation 2.1) does not have the Q term (Q = 0). The element stiffness matrix is the matrix that multiplies the column vector of nodal values, $\{C^{(e)}\}$. While the vector {R} represents a system of equations, the final result is a system of first-order differential equations given by

$$\{R\} = [K]\{C\} + [P]\{\overline{C}\} = \{0\} \tag{4.6}$$

where the boundary conditions have been incorporated into [K] and {F} and the interelement requirements discarded.

The vector $\{\overline{C}\}$ is $\{\overline{C}\}^T = \left[\frac{\partial C_1}{\partial t}, \frac{\partial C_2}{\partial t}, \frac{\partial C_3}{\partial t}, ,,,,, \frac{\partial C_n}{\partial t}\right]$ (4.7)

The matrix [P] is usually called the capacitance matrix, which is:

$$[p^{(e)}] = \frac{\lambda L}{2}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (4.8)$$

for the one-dimensional linear element. And $[p^{(e)}]$ is combined with the other element matrices and summed over all the elements using the direct stiffness procedure.

The finite element solution of time-dependent field problems produces a system of linear first-order differential equations in the time domain. These equations must be solved before the variation of C in space and time is known. There are several procedures for numerically solving the equations given by Equation 4.6.

The mean value theorem for differentiation to develop an equation for a given function $f(t)$ and the time interval [a, b] and the central difference method (θ = 0.5) give the following equation;

$$\left([P] + \frac{\Delta t}{2}[K]\right)\{C\}_b = \left([P] - \frac{\Delta t}{2}[K]\right)\{C\}_a \quad (4.9)$$

where $\{C\}_a$ and $\{C\}_b$ are the concentrations of the contaminant at the nodes at certain times a and b, and $\Delta t = b - a$.

Equation 4.9 is the numerical solution of the system of differential equation for the time-dependent diffusion problem. The $\{C\}_a$ and $\{C\}_b$ values in Equation 4.9 will be changed for each time increment. However, the time

interval must be selected so as to avoid the failure of the calculated values to satisfy physical requirements and the problem of numerical oscillations. For the one-dimensional lumped formulation, the time interval should satisfy the following relationship:

$$\Delta t < \frac{\lambda L^2}{4D(1-\theta)} \tag{4.10}$$

where,

λ : *Constant (1 for the diffusion problem)*

L : *Thickness of polymer, cm*

D : *Diffusion Coefficient, cm^2/sec*

θ : *Ratio of the time intervals (0.5 for the central difference method)*

4.3. Computer Programming

The equations resulting from the finite element method cannot be easily solved manually, since the number of equations is usually very large; however, the finite element method can be easily coded into a computer program.

The computer program, named "MigFem 2001" for this study, was developed using the Visual Basic (Microsoft™) program language. The general computer program, developed by Segerlind (1984) using the "FORTRAN" and "QBASIC" programming languages, for solving a partial differential equation in one dimension with the finite element method, was modified to be flexible enough to fit the physical problem in this study, diffusion and migration, and the user interface for the "WINDOWS" application was developed solely for this study. Program codes are attached in Appendix B.

The program consists of 3 "Forms" and 4 "Modules"; "Forms" are "frmSplash", "frmMain", and "frminput5" and "Modules" are "dimModule5", "BandMtx5", "TimeMtx5", and "Output5".

The "frmSplash" controls a program-opening window which is splashed as opening the program (Figure 4.2); the "frmMain" functions as a tool to handle the basic "WINDOWS" components such as "open", "save", "save as", etc.; and the "frminput5" is the main program to input and process the data and calculate and present the results (Figures 4.3 - 4.6).

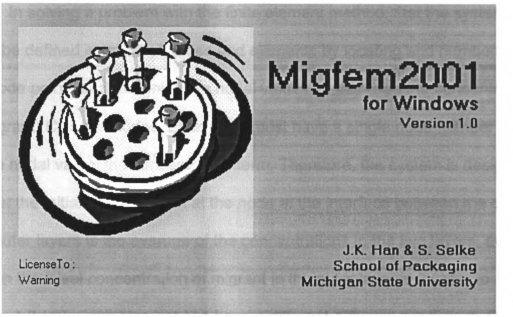


Figure 4.2 Graphic object for program opening: "frmSplash"

For the "Modules", the "dimModule5" specifies the dimensions of the variables; the "BandMtx5" contains the subprograms related to the modification and solution of a system of equations; the "TimeMtx5" contains the subprogram related to a finite element analysis in time; and "Output5" places the values in final solutions.

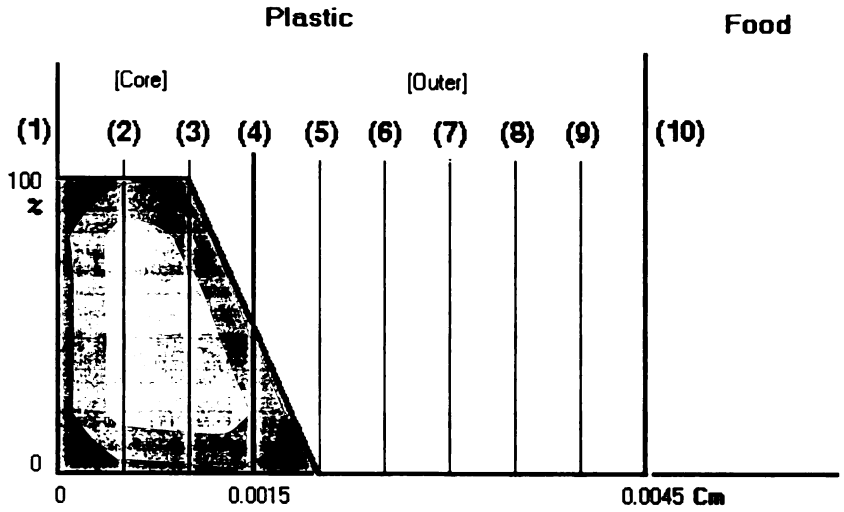
The "MigFem 2001" program requires the input of several parameters and by clicking an appropriate "Command Button," calculations and simulations can be readily performed. In this chapter, the basic operation of the program and handling of the parameters are explained briefly, and detailed applications of the program will be explained in the "Results and Discussion" section.

4.3.1. Concept of FEM Application to Migration Problems

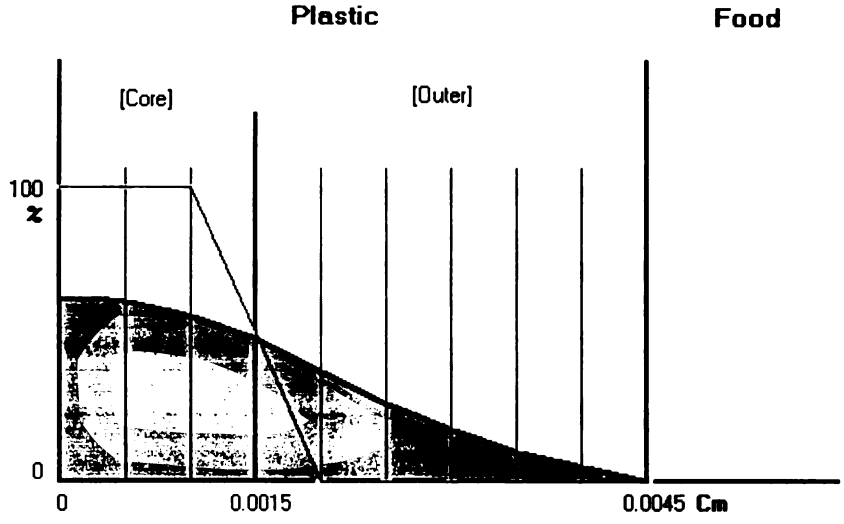
In solving a problem with the finite element method, first the system must be defined in terms of nodes and elements by locating and numbering the node points and assigning the nodal values, in this case the concentration of migrant, *C*, at the nodes. Each node must have a single node number and single nodal value, the concentration term. Therefore, the system is designed so that the initial concentration at the node at the interface between the core and outer layers is the average of the concentrations in the two layers, even though the initial concentration of migrant in the outer layer is 0 (refer to Figure 4.3 (a)). Each element can have a different diffusion coefficient, *D*.

Figure 4.3 (a) illustrates the nodes, elements, and nodal values. The numbers enclosed in parentheses represent the nodes; the spaces between the nodes are the elements; and relative concentrations are assigned for all nodes.

The shaded area represents the initial amount of the migrant in the polymer at $t = 0$. Figure 4.3 (b) shows a concentration distribution profile of the migrant after a certain period of time ($t = t$), with the shaded area again representing the amount of the migrant in the polymer. The amount of migrant migrated into the food can be determined by calculating the difference between the amounts of shaded area at $t = 0$ and at $t = t$.



(a)



(b)

Figure 4.3 Concepts of the finite element method: (a) initial condition (b) distribution of migrant after a period of time

4.3.2. Basic Operation of "MigFem 2001"

The visual interface of the program consists of 4 tabs: "System Design", "Experiment", "Calculation", and "Simulation." Operation methods and functions of each component or object under the tabs are as follows:

4.3.2.1. "System Design" Tab

Figure 4.4 shows one of the graphic objects, under the "System Design" tab in the form "frminput5" of the program. Some basic components of the finite element can be designed and appropriate values should be entered.

Frame "Thickness of Layer"

The finite element method has no limitation on the number of layers of the multilayer structures; however, the program was developed to calculate migration in up to 3 layers, for practical reasons. Thicknesses of up to three layers can be entered and by clicking the command button, "Total thickness," total thickness of interest will be calculated and presented in the appropriate text boxes.

Frame "FEM Analysis Data"

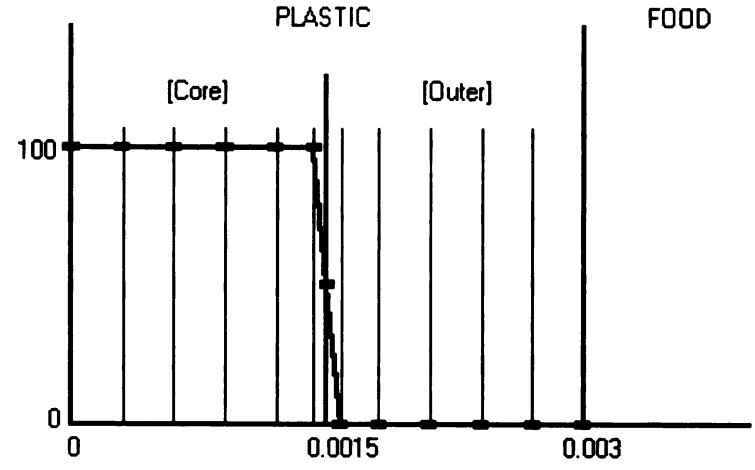
The number of layers should be entered in the text box. The thickness of a structure of interest should be divided into a finite number of elements for calculation. The program was developed to have up to 14 elements (15 nodes), for practical reasons. Dr. Segerlind, author of "Applied Finite Element

Analysis" (Segerlind, 1984) recommended about 15 – 20 elements for the particular physical problem in this study, and also he stressed the need for using a unit grid, in which every element has the same length or distance, to reduce the round-up error and/or numerical oscillations from the FEM calculation. It is possible to make the length of interface elements shorter than the other elements to describe a realistic distribution of contaminants in the polymer (Figure 4.5 (a)); however, the calculation result by the FEM shows negligible difference from the unit grid approach (Figure 4.5 (b)). The variable grid approach requires an adjustment of Δt for the reduced element length to avoid round-up errors and/or oscillations. The unit grid approach was used in this study since in practice there was no difference between the two approaches in the calculations by the FEM.

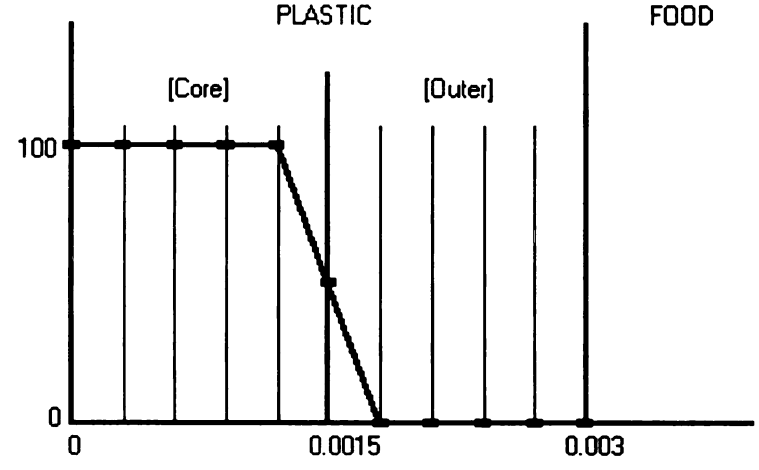
After the number of elements is entered in the text box, by clicking the command button, "Grid, Xi", coordinate values (thickness terms) will be calculated and presented in the frame "Element Node Data".

From the assumptions for modeling, there is a constant zero concentration of the contaminant at the polymer/food interface. Therefore, "Number of Known Conc.", which stands for the number of nodes whose concentrations are already known, should be 1 and "Nodes #" will be the number of the node which represents the interface of the polymer and food or food simulant.

Figure 4.4 Graphic object under "System Design" tab



(a)



(b)

Figure 4.5 Design of the grid in the FEM: (a) variable grid (b) unit grid

Frame "Element Node Data"

"Coordinate" will be presented automatically by clicking "Grid, Xi", or can be input manually. "Initial Value" represents the initial distribution of contaminant concentrations. A relative concentration or percent will be used for the FEM calculation as explained above, and the details will be discussed in the "Results and Discussion" section. For the text boxes under "Layers", the number of the layers corresponding to the element numbers should be entered. After that, clicking the command button, "Check System" will show the designed system graphically for the finite element analysis.

4.3.2.2. "Experiment" Tab

Figure 4.6 shows one of the graphic objects, under the "Experiment" tab in the program. Experimental results will be entered under this tab.

Frame "Coefficients"

Diffusion coefficients of the contaminant in each layer of multilayer structures and the partition coefficient of the contaminant between the polymer (outer layer) and the food should be entered.

Frame "Polymer"

Densities of each polymer layer of multilayer structures and the surface area of the structure contacting the food should be entered. Clicking the command button, "Weight, g", will calculate the total weight and volume of multilayer structures.

Frame "Food/Liquid"

The volume and density of the food contacting the multilayer structure should be entered. Clicking the command button, "Weight, g", will calculate the total weight of the food.

Frame "Migrant"

Initial concentration of the contaminant or migrant in the core layer should be entered. Clicking the command button, "Final Conc. in Food", will calculate the concentration of contaminant in the food after complete migration.

Frame "Diffusivity by ASTM D4754"

The diffusion coefficient can be determined by the two-sided liquid extraction method for plastic materials (ASTM D 4754) and Equation 2.7. A simple program for calculating the diffusion coefficient was included in the program. The parameters to be entered are the following:

- Migration data: time in $hr^{1/2}$ vs. C_t (concentration of migrant in the food, ppm, $\mu g/cm^3$)
- Initial concentration of the migrant in the polymer, ppm ($\mu g/g$)
- Density of polymer, g/cm^3
- One half thickness of polymer, cm
- Volume of the food/liquid, cm^3
- Surface area of polymer contacting the food/liquid, cm^2

Figure 4.6 Graphic object under “Experiment” tab

After entering all the data, by clicking the command button, "Plot", extraction data as a function of the square root of time will be plotted. The number of data sets within a straight line may be determined visually. Selecting "No. of Data Set" and clicking the command button, "RUN", will calculate the diffusion coefficient in cm^2/sec.

4.3.2.3. "Calculation" Tab

Figure 4.7 shows one of the graphic objects, under the "Calculation" tab in the program. The time interval, Δt, will be entered and two options for calculating the amount of migration will be provided.

Frame "Delta Time"

The time interval, Δt, must be selected so as to avoid the failure of the calculated values to satisfy physical requirements and the problem of numerical oscillations. In this program, the time interval should satisfy the conditions of Equation 4.10. The command button, "INPUT" should be clicked to proceed to the next step, "Calculation".

Frame "Calculation"

There are two options for calculating the migration of contaminant into the food. By selecting the option button, "By Steps", migration of contaminant to the food can be calculated for accumulated time intervals as the command button, "Next", is clicked continuously. Selecting the option button, "By

Hours", and entering a time value in hours can calculate migration of the contaminant at the time of interest.

4.3.2.4. "Simulation" Tab

Figure 4.8 shows one of the graphic objects, under the "Simulation" tab in the program. The amount of migration can be plotted as a function of time and compared with actual test results.

Frame "Data from Migration Test"

If there are data from actual migration tests, they can be entered in the frame for the purpose of comparison with the simulation results. Concentrations of the migrant (ppm, $\mu g/cm^3$) in the food as a function of square root of time should be entered.

Graphical Presentation of Simulation Results

Entering a period of time (in hours) of interest and clicking the command button, "Simulation", will show the plot of the relative migration, M_t/M_e, as a function of the square root of time in a graphic area. To compare the migration test results and simulation results, the number of data sets from the frame, "Data From Migration Test", should be entered before clicking the command button, "Test & Simulation".

Figure 4.7 Graphic object under "Calculation" tab

Figure 4.8 Graphic object under “Simulation” tab

5. RESULTS AND DISCUSSION

5.1. Standard Calibration Curve for HPLC

The chromatogram from HPLC was calibrated with the external calibrating method by using standard solutions of the contaminants in appropriate food simulant solutions. The solutions were prepared by varying the concentrations of the contaminant in appropriate solutions: 10, 25, 50, 75, and 100 ppm (μg/ml).

A 20 μl sample was removed from each of the standard solutions and injected to the HPLC. The standard curve of the area responses for each contaminant concentration was constructed and the calibration factor was determined from the slope of the linear plot.

The standard calibration curves were made regularly during the experiments, and representative curves for contaminants with specific solutions, BHT with acetonitrile, 100% ethanol, and 50% aqueous ethanol and Irganox™1076 with methylene chloride and 100% ethanol are presented in Appendix A.

5.2. Characterization of the Model Structures

5.2.1. Thickness of the Layers and Weight of Polymer Samples

The thicknesses of the core layer and outer layer are among the most important parameters in modeling migration through multilayer polymer structures. The thicknesses of each layer were controlled precisely by automatic devices during extrusion. Table 5.1 shows the results of thickness measurements, which are in good agreement with the designed thicknesses of the samples. Table 5.2 shows the results of weight measurements of polymer coupon samples for migration and partition experiments.

5.2.2. Initial Concentration of Contaminants in the Core Layer

Eventually, the total amount of contaminant migrated to the food is determined by the initial concentration of contaminant in the core layer. Migration tests in this study were performed at exaggerated conditions in order to obtain high and precise migration values which are essential for comparing with simulation data. Therefore, polymer samples with extremely high initial concentrations of the contaminant simulants were made.

To determine the initial concentration of contaminant simulant in the polymer sample, the Soxhlet extraction method for thinner (up to 2.4 mil) samples and the migration cell extraction method for thicker samples were used. Initial concentrations of the contaminant simulants are presented in Table 5.3 for BHT and Table 5.4 for Irganox™1076. To validate the extraction

methods, both methods were used to determine the initial concentration of contaminants in thinner samples, and the results, which show good agreement, are included in Table 5.3 and Table 5.4

The initial concentrations, measured by the migration cell method, were used in the model; however, these values were converted to the maximum possible concentrations in the food simulants after complete extraction/migration from 20 polymer sample coupons (the amount used in the migration test). These values, included in the last column of Table 5.3 and Table 5.4, will be regarded as the initial concentration of the contaminant simulants, expressed as M_e, in the polymer samples so that the relative or % migration can be calculated. With these initial concentration values, the amount of contaminant migration into the food simulants at certain periods of time can be easily expressed as percent (%) of the maximum concentration in the food simulants (complete migration).

5.2.3. Volume of the Food Simulants

The food simulant was added to the glass vial migration cell to its shoulder, a total of 36 ml, which was the same for both migration and partition experiments.

Table 5.1 Designed and measured thickness of polymer samples

No.	Designed Structure			Measured total thickness, mil
	Construction	Thickness of layers, mil	Total thickness, mil	
1	Single layer with core* material	1.2	1.2	1.20
2	HDPE/Core*/HDPE	0.6/1.2/0.6	2.4	2.40
3	HDPE/Core*/HDPE	1.2/1.2/1.2	3.6	3.60
4	HDPE/Core*/HDPE	1.8/1.2/1.8	4.8	4.90
5	HDPE single layer**	1.2	1.2	1.17

* The core layer consists 10 % LDPE and 90 % HDPE (w/w)
** HDPE single layer without the contaminant simulant was used in partition experiment.

Table 5.2 Weight of polymer coupon samples

No.	Construction	Total thickness, mil	Weight of coupon, g	Surface area of coupon*, cm^2
1	Single layer with core material	1.20	0.0107	3.6305
2	HDPE/Core/HDPE	2.40	0.0216	3.6305
3	HDPE/Core/HDPE	3.60	0.0323	3.6305
4	HDPE/Core/HDPE	4.90	0.0428	3.6305
5	HDPE single layer	1.17	0.0104	3.6305

* Calculated surface area of single side of a coupon.

Table 5.3 Initial concentration of BHT in the core layer of polymer samples

No.	Construction	Total thickness, mil	Initial concentration, ppm (w/w)		Maximum conc.* in food, ppm
			Soxhlet	Migration Cell	
1	Single layer with core material	1.20	9,220	9,648	55.93
2	HDPE/Core/HDPE	2.40	13,060	13,560	78.60
3	HDPE/Core/HDPE	3.60	n/a	15,048	87.23
4	HDPE/Core/HDPE	4.90	n/a	14,470	83.88
5	HDPE single layer	1.17	n/a	n/a	n/a

* Concentration of contaminants in the food simulant after complete migration from 20 sample disks.

Table 5.4 Initial concentration of Irganox™1076 in the core layer of polymer samples

No.	Construction	Total thickness, mil	Initial concentration, ppm (w/w)		Maximum conc.* in food, ppm
			Soxhlet	Migration Cell	
1	Single layer with core material	1.20	10,918	11,312	65.57
2	HDPE/Core/HDPE	2.40	12,545	13,034	75.55
3	HDPE/Core/HDPE	3.60	n/a	12,540	72.69
4	HDPE/Core/HDPE	4.90	n/a	11,462	66.44
5	HDPE single layer	1.17	n/a	n/a	n/a

* Concentration of contaminants in the food simulant after complete migration from 20 sample disks.

5.2.4. Diffusion Coefficients of Contaminants through the Polymer Layers, Core and Outer layers

The diffusion coefficients of the contaminant in each layer of multilayer structures are not only the most important parameters to determine the rate of migration into the food or food simulant, but also the critical factors for comparing the computer model with the actual results.

A single layer of core material (10% LDPE and 90% HDPE) with a known amount of contaminant simulant, BHT or Irganox™1076, was readily available so that it was possible to determine the diffusion coefficients of contaminant simulants in the core layer with the analytical methods described above and Equation 2.7. However, the diffusion coefficients of the contaminant simulants, BHT and Irganox™1076, in the outer layer were estimated algebraically using empirical equations, since the needed sample, a single outer layer with a known amount of contaminant, was not available.

5.2.4.1. Diffusion Coefficients of Contaminants through Core Layer

The experimental data for migration of BHT and Irganox™1076 into 100% ethanol correlate well with Equation 2.7. The plot of extraction data up to 50 – 60 % migration as a function of the square root of time showed a straight line as shown in Figures 5.1 and 5.2. The slope of the line was used to determine the diffusion coefficients of contaminant simulants, BHT and Irganox™1076, in the core layer.

Applying Equation 2.7 to the data at each storage temperature (23, 31,

and 40°C), the following diffusion coefficients were calculated: for BHT, 2.34 x 10^{-11} cm^2/sec at 23°C, 8.65 x 10^{-11} cm^2/sec at 31°C and 2.20 x 10^{-10} cm^2/sec at 40°C; for Irganox™1076, 1.38 x 10^{-12} cm^2/sec at 23°C, 6.58 x 10^{-12} cm^2/sec at31°C and 3.04 x 10^{-11} cm^2/sec at 40°C. The measured diffusion coefficients are summarized in Tables 5.5 and 5.6, and plotted in an Arrhenius relationship in Figures 5.3 and 5.4; showing a linear relationship between the log value of the diffusion coefficient and the inverse of the absolute temperature.

The experimental data for migration of BHT into 50% ethanol also correlate well with Equation 2.7 and diffusion coefficients may be calculated; however, a diffusion coefficient obtained with these data only represents an effective or apparent diffusion coefficient of a specific system, i.e. BHT into 50% ethanol from HDPE. A few previous studies (Schwope et al 1987, Lickly et al 1990, and Linssen et al 1998) also presented a diffusion coefficient value as an effective or apparent diffusion coefficient for a specific environment.

A diffusion coefficient should be an intrinsic physical and chemical property of a contaminant simulant in a polymer and can be regarded as only dependent on temperature. The slower migration or lower apparent diffusion coefficient of contaminants into 50% ethanol can be explained by introducing the concept of partition between the phases. Therefore, identical diffusion coefficient values were applied in modeling both cases: migration to 100% ethanol and to 50% ethanol.

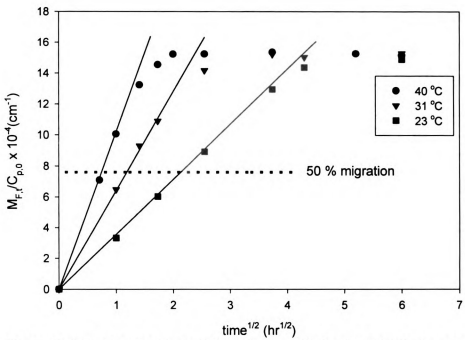


Figure 5.1 Determination of diffusion coefficients of BHT in core layer

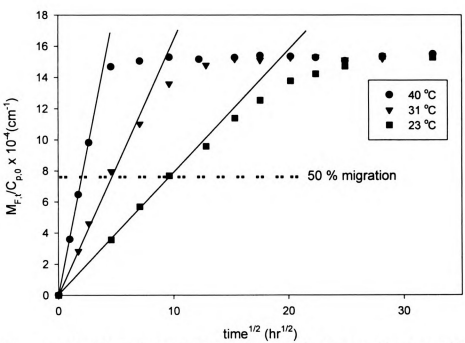


Figure 5.2 Determination of diffusion coefficients of Irganox™1076 in core layer

Table 5.5 Diffusion coefficients of BHT in the core layer measured with 100% ethanol

Temperature, °C	Composition of Core Layer	Diffusion Coefficient, D, cm^2/sec
23	10% LDPE / 90% HDPE	2.34×10^{-11}
31	10% LDPE / 90% HDPE	8.65×10^{-11}
40	10% LDPE / 90% HDPE	2.20×10^{-10}

Table 5.6 Diffusion coefficients of Irganox™1076 in the core layer measured with 100% ethanol

Temperature, °C	Composition of Core Layer	Diffusion Coefficient, D, cm^2/sec
23	10% LDPE / 90% HDPE	1.38×10^{-12}
31	10% LDPE / 90% HDPE	6.58×10^{-12}
40	10% LDPE / 90% HDPE	3.04×10^{-11}

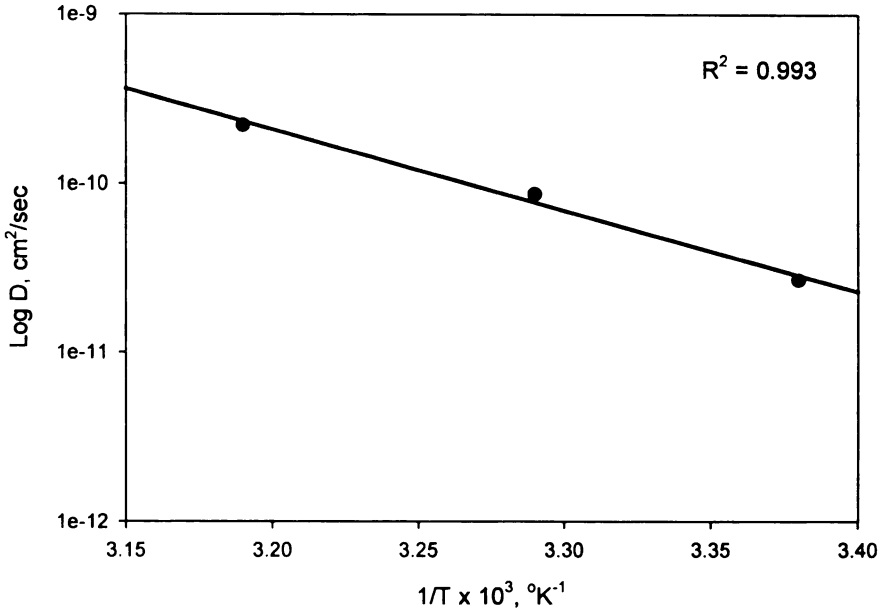


Figure 5.3 Arrhenius plot of diffusion coefficient of BHT in core layer in temperature range of 23°C – 40°C

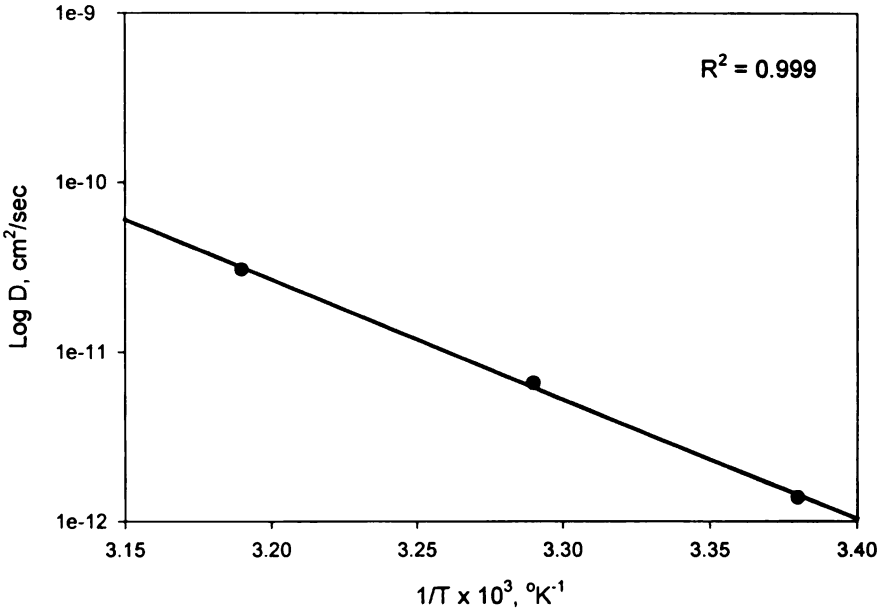


Figure 5.4 Arrhenius plot of diffusion coefficient of Irganox™1076 in core layer in temperature range of 23°C – 40°C

5.2.4.2. Diffusion Coefficients of Contaminants through Outer Layer

While the core layer was a blend of 10% LDPE and 90% HDPE, the outer layer was made of 100% HDPE, identical to that used in the core layer. The diffusion coefficient of the contaminant in the outer layer was estimated algebraically using an empirical equation (Baner et al. 1994, 1996; Piringer 1994).

These authors developed the following relationship between the diffusion coefficient and temperature and the migrant's molecular weight:

$$D_p \leq 10{,}000 \cdot \exp\left(A_p - aM_r - b\frac{1}{T} \right) \qquad (5.1)$$

where A_p is a dimensionless polymer-specific constant; a and b are constants accounting for the dependence of diffusion on the migrant's molecular weight and temperature; M_r is the molecular weight of the migrants; and T is the temperature in Kelvin.

Baner et al. and Piringer determined the value of the coefficients a and b as 0.01 and 10450 and presented some values of A_p for polyolefin polymers (Table 5.7). This enables estimation of the highest value of migration and estimated diffusion coefficient values will thus lead to conservative migration estimates through likely over-estimation of the diffusion coefficients. The diffusion coefficient estimated by Equation 5.1 will not be used for modeling in this study; however, a relationship between the diffusion coefficient of LDPE and HDPE can be utilized to estimate the diffusion coefficient in the outer layer (100% HDPE) of the sample.

Reynier et al. (1999) later proposed improved and modified values of the constant A_p in the above equation, applying experimental data determined by a film to film diffusion method; these are included in Table 5.7. These A_p values may be more suitable for estimating the diffusion coefficients of migrants of a low molecular weight such as BHT.

The approach by Baner et al. (1994, 1996) for calculating diffusion coefficients results in a relative value of the diffusion coefficient for any migrant in LDPE that is 20 times higher than that of HDPE. The A_p proposed by Reynier et al. calculates a diffusion coefficient for LDPE that is only 3 times higher than that of HDPE. One A_p value out of the two different approaches should be selected to estimate the diffusion coefficient of the contaminants in the outer layer appropriately.

The National Bureau of Standards (now the National Institute of Standards and Technology) collected migration data under contract to FDA (NBSIR 82-2472). The final report includes a table of apparent diffusion coefficients determined from data for migration of various ^{14}C-labelled migrants from polyolefin polymers to various liquid food simulants. From the table, it can be seen that the difference in diffusion coefficient between LDPE and HDPE for lower molecular weight migrants such as n-octadecane (254.5 daltons) and BHT (221.0 daltons) is much smaller than the difference in diffusion coefficient for higher molecular weight migrants such as n-dotriacontane (450.9 daltons). In this study, the A_p values by Reynier et al.

were used for BHT and the A_p values by Baner et al. were used for Irganox™1076 to calculate the diffusion coefficients in the outer layer.

By using the weighted average calculation method and A_p values presented in Table 5.7, the following simultaneous equations can be written for the diffusion coefficients of the contaminant simulants, BHT and Irganox™0176 in LDPE and HDPE at 40 °C;

For BHT,

$$0.1\ D_{LDPE} + 0.9\ D_{HDPE} = 2.20 \times 10^{-10}\ cm^2/sec$$

$$D_{LDPE} = 3D_{HDPE}$$

For Irganox™1076,

$$0.1\ D_{LDPE} + 0.9\ D_{HDPE} = 3.04 \times 10^{-11}\ cm^2/sec$$

$$D_{LDPE} = 20D_{HDPE}$$

where D is the diffusion coefficient.

By solving the equations simultaneously, the relationship of the diffusion coefficients in the core and outer layers can be determined. The diffusion coefficient of BHT and Irganox™1076 in the outer layer (100% HDPE) is about 20% and 65% less than the diffusion coefficient in the core layer (10% LDPE + 90% LDPE). The estimated diffusion coefficients of BHT in the outer layer at various temperatures are presented in Tables 5.8 and 5.9.

Table 5.7 Estimation of constants, A_p of Equation 5.1

Authors	LDPE	HDPE	PP
Baner et al.	11	8	6
Reynier et al.	9	8	6.5

Table 5.8 Estimated diffusion coefficients of BHT in the outer layer at different temperatures.

Temperature, °C	Composition of Outer Layer	Diffusion Coefficient, D, cm^2/sec
23	100% HDPE	1.87×10^{-11}
31	100% HDPE	6.92×10^{-11}
40	100% HDPE	1.76×10^{-10}

Table 5.9 Estimated diffusion coefficients of Irganox™1076 in the outer layer at different temperatures.

Temperature, °C	Composition of Outer Layer	Diffusion Coefficient, D, cm^2/sec
23	100% HDPE	4.83×10^{-13}
31	100% HDPE	2.30×10^{-12}
40	100% HDPE	1.07×10^{-11}

5.2.5. Partition Coefficient for the Contaminant between the Outer Layer and the Selected Food Simulants

5.2.5.1. Measurement of Equilibrium Partition Coefficient

The migration cell extraction method with the polymer sample without contaminant (0% contaminant) and HPLC as an analytical method were used to determine the partition coefficients for the HDPE/BHT/food simulant systems used.

The extraction cells with food simulants containing 100 ppm (g/cc) of contaminant simulants were stored at 23°C, 31°C, and 40°C until the migration process (reverse direction mass transfer from food to polymer) reached equilibrium. The storage time needed for the migration process to reach equilibrium for the HDPE/BHT/100% ethanol system and the HDPE/BHT/50% ethanol system was more than 60 hours at 23°C, 20 hours at 31°C, and 8 hours at 40°C.

After the migration process reached equilibrium, both the food simulants and the plastics were analyzed for contaminant simulant content, and partition coefficients were calculated by Equation 3.2 for each sample. Results are summarized in Table 5.9 and the Arrhenius relationship is presented in Figure 5.5.

Table 5.10 Equilibrium partition coefficient of BHT between HDPE selected food simulants at different temperatures.

Food Simulant	Temperature, °C	Partition Coefficient, K_L^P, $^{g/g}/_{g/g}$
100% Ethanol	23	0
	31	0
	40	0
50% Ethanol	23	23.76
	31	19.77
	40	12.75

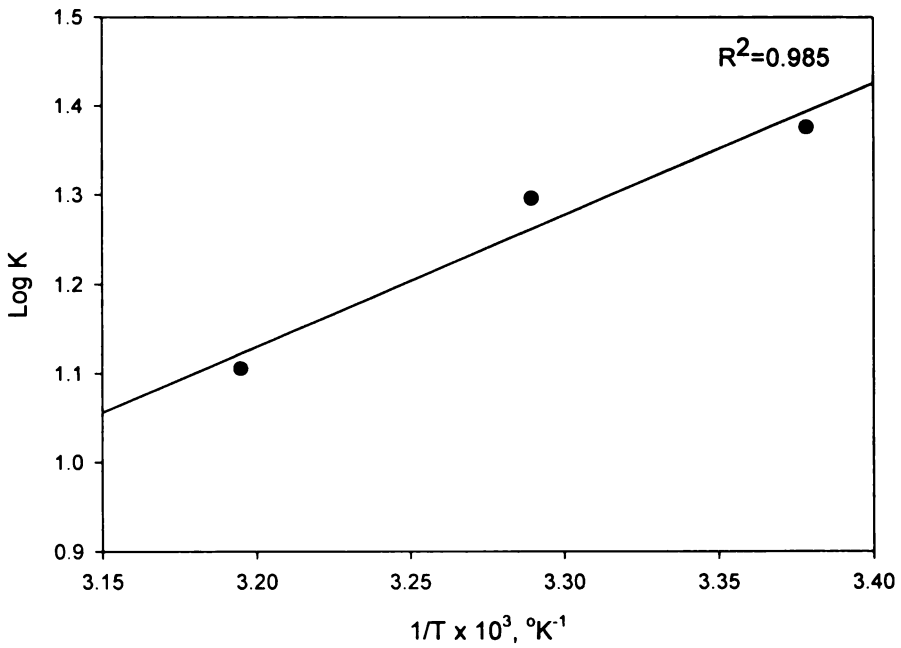


Figure 5.5 Arrhenius plot of partition coefficient of BHT between HDPE and 50% ethanol in temperature range of 23°C – 40°C

5.2.5.2. Determination of Percent Partition for Change in Thickness of Outer Layer

The partition coefficient, *K*, is defined by the ratio of the concentrations of a substance in the polymer and in the contact liquid (food simulant) at equilibrium. It characterizes the partition equilibrium of the substance (contaminant simulant in this study) between the polymer and food simulant, and its value provides a measurement of the maximum extent of migration or sorption potential of the substance into the food or polymer.

To incorporate the partition coefficient in the computational model, it was converted to a term defined as percent partition, PP, relating the mass or volume of the polymer and the food simulant and the masses of the contaminant in the polymer and in the food simulant for a specific system.

The following relationship results from the mass balance of partitioning of a substance between the polymer (P) and the food simulant (L) at equilibrium:

$$\frac{m_{L,e}}{m_{P,o}} \times 100 = \frac{1}{1+\alpha} \times 100 = PP \tag{5.2}$$

with α defined as

$$\alpha = K_L^P \left(\frac{W_P}{W_L} \right) \tag{5.3}$$

where,

PP : *Percent partition*

$m_{L,e}$: *Mass of contaminant in the food simulant at equilibrium with partitioning, g*

$m_{P,O}$: *Mass of contaminant in the polymer at initial time or mass of contaminant when completely migrated into the food simulant, g*

K_L^P : *Partition coefficient*

W_P : *Mass of the polymer, g*

W_L : *Mass of the food simulant, g*

It was assumed that the partition coefficient is independent of time (Crank 1975, Vergnaud 1992). The amount of contaminant migrated to the food at time t now can be estimated with the following relationship:

$$m_{L,t} = {}^{PP}\!/_{100} \times m_{m,t} \qquad (5.4)$$

where,

$m_{L,t,}$: *Mass of contaminant migrated to the food at time t with partitioning, g*

$m_{m,t}$: *Mass of contaminant migrated to the food at time t without partitioning, g*

The finite element method used in the computational model calculates the migration amount, $m_{m,t}$, from the polymer to the food simulant, without partitioning repeatedly with increasing small time increments Δt. Migration of

contaminant to the food simulant with partitioning, $m_{L,t,}$, can then be effectively estimated by using Equation 5.4.

For the functional barrier polymer system, the percent partition must be adjusted for the mass (or volume) of the outer layer, which is free from contaminant at the initial time. Addition of the outer layer to the core layer, which already has a certain amount of contaminant, will change the total mass of the polymer and therefore alter the partition mass balance.

The percent partition, PP, of the HDPE/HBT/50% ethanol system for different thicknesses of the outer layer and different temperatures is summarized in Table 5.11.

Table 5.11 Percent partition of the HDPE/BHT/50% ethanol system for different thicknesses of outer layer and temperature

Temp. °C	Partition coefficient, K_L^P	Effective thickness*, mil (core/outer)	Effective thickness* ratio	Percent partition, *PP*
23	23.76	0.6/0.0	1 : 0	86.3
		0.6/0.6	1 : 1	75.5
		0.6/1.2	1 : 2	67.3
		0.6/1.8	1 : 3	60.7
31	19.77	0.6/0.0	1 : 0	88.3
		0.6/0.6	1 : 1	78.8
		0.6/1.2	1 : 2	71.2
		0.6/1.8	1 : 3	64.9
40	12.75	0.6/0.0	1 : 0	92.2
		0.6/0.6	1 : 1	85.2
		0.6/1.2	1 : 2	79.3
		0.6/1.8	1 : 3	74.2

*For the double side extraction method, the effective thickness is one half of the core layer and the full thickness of one outer layer.

5.3 Analysis of Migration Tests

5.3.1. Kinetics of BHT Migration from the Polymer to the Selected Food Simulants

According to Fickian diffusion theory, if diffusion is the controlling mechanism in migration, the amount of contaminant which migrates from a polymer is proportional to the square root of time. The relative migration, M_t/M_e can be presented graphically as a function of the square root of time; this has the advantage that it allows identification of whether the migration process is diffusion-controlled, Fickian. If so, modeling based on Fick's 2nd equation is valid.

Figure 5.6 shows the relative migration values, M_t/M_e, of BHT from the core layer polymer material into 100% ethanol as a function of the square root of time at various temperatures. The graphs show good agreement with Fickian theory (refer to Figure 5.2 for Irganox™1076). At each temperature, migration data yield a straight line until about 50 - 60% of BHT has migrated, at which point the depletion of BHT in the polymer becomes a factor. As expected, temperature plays an important role in the rate of the migration. Regardless of temperature, BHT was completely migrated into 100% ethanol and neither a partitioning effect nor degradation of BHT in ethanol was observed.

Figure 5.7 shows the relative migration values, M_t/M_e, of BHT from the core layer polymer material into 50% aqueous ethanol. The graphs also show

fairly good agreement with Fickian theory. Several previous studies had calculated a diffusion coefficient, the so-called effective or apparent diffusion coefficient, with this relationship.

Temperature was an important factor in determining the rate of migration into 50% aqueous ethanol. Apparent equilibrium partitioning effects were observed at all temperatures. Partition coefficients were determined by the migration cell absorption method in this study and they were converted to percent partition for the single core layer polymer: 92.2 at 40°C, 88.3 at 31°C, and 86.3 at 23°C. These percent partition values are a little higher than the actual experimental value, M_t/M_∞ value at equilibrium in Figure 5.7. The differences between the percent partition and the experimental value might result from the difference in polymer materials: 100% HDPE for determination of the partition percent and 10% LDPE + 90% HDPE for the actual experiment. Gandek et al (1998) pointed out that the partition coefficient is a weak function of temperature. It was observed that the dependence of the partition coefficient on temperature appeared to be much smaller than the dependence of the diffusion coefficient on temperature in this study. However, the temperature dependence of the partition coefficient was large enough to be included in the modeling.

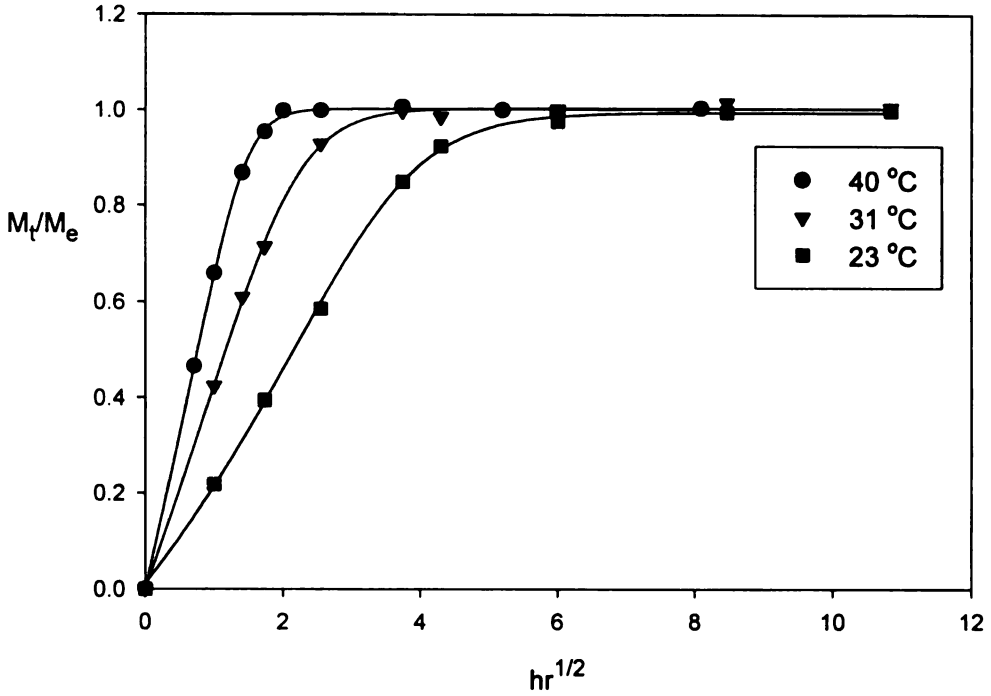


Figure 5.6 Relative migration, M_t/M_e, of BHT from the core layer to 100% ethanol as a function of time and various temperatures

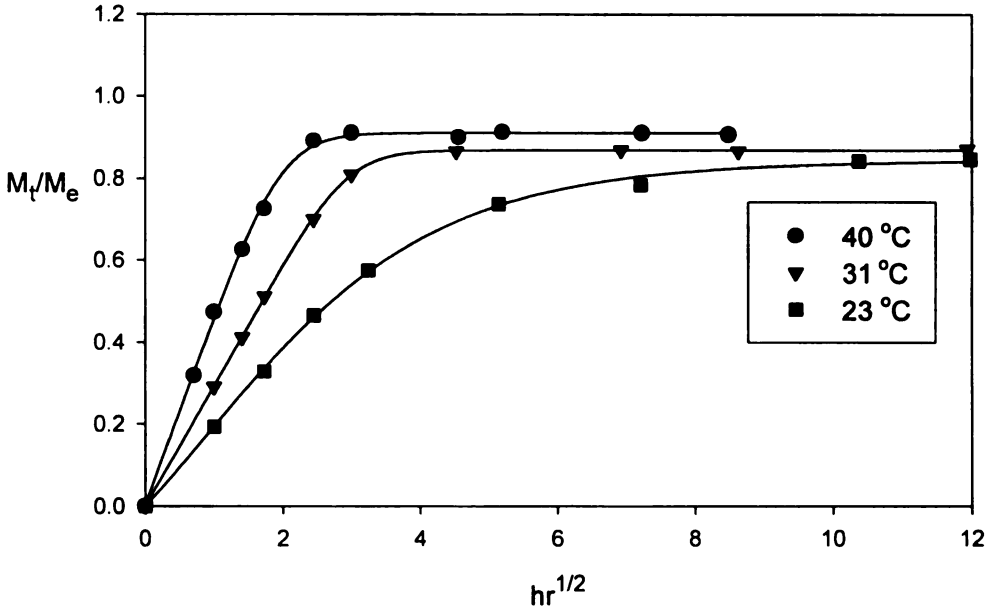


Figure 5.7 Relative migration, M_t/M_e, of BHT from the core layer to 50% ethanol as a function of time and various temperatures

5.3.2. Effect of Thickness of Functional Barrier on Migration

Figures 5.8 to 5.10 show migration of BHT as a function of time from the core layer through various thicknesses of outer layer (functional barrier) to 100% ethanol, and Figures 5.11 to 5.13 show migration of BHT as a function of time to 50% aqueous ethanol at test temperatures of 23, 31, and 40°C. Even though complete migration was not accomplished in the 4 months of the experiment, migration results of Irganox™1076 to 100% ethanol as a function of time are presented in Figures 5.14 to 5.16.

As expected, the rate of migration decreased with increasing thickness of the outer layer. The storage time needed for the migration process to reach equilibrium for the HDPE/BHT/ethanol system was extended substantially with increasing thickness of the outer layer. Especially at early time periods, for example less than 4 hours at 40°C, a lag time effect due to the contaminant-free functional barrier layer was observed for all three-layer structures (Figures 5.8 and 5.11).

The storage times for equilibrium (over 99% migration) of BHT migration for each temperature and sample structure are presented in Table 5.12.

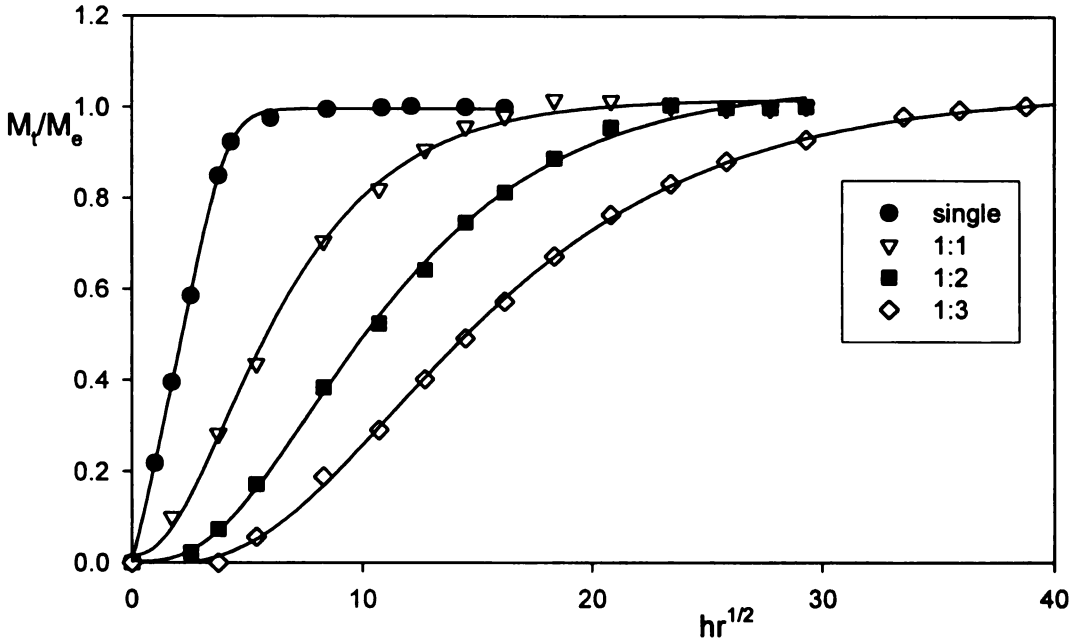


Figure 5.8 Effect of thickness of functional barrier on migration rate of BHT to 100% ethanol at 23°C

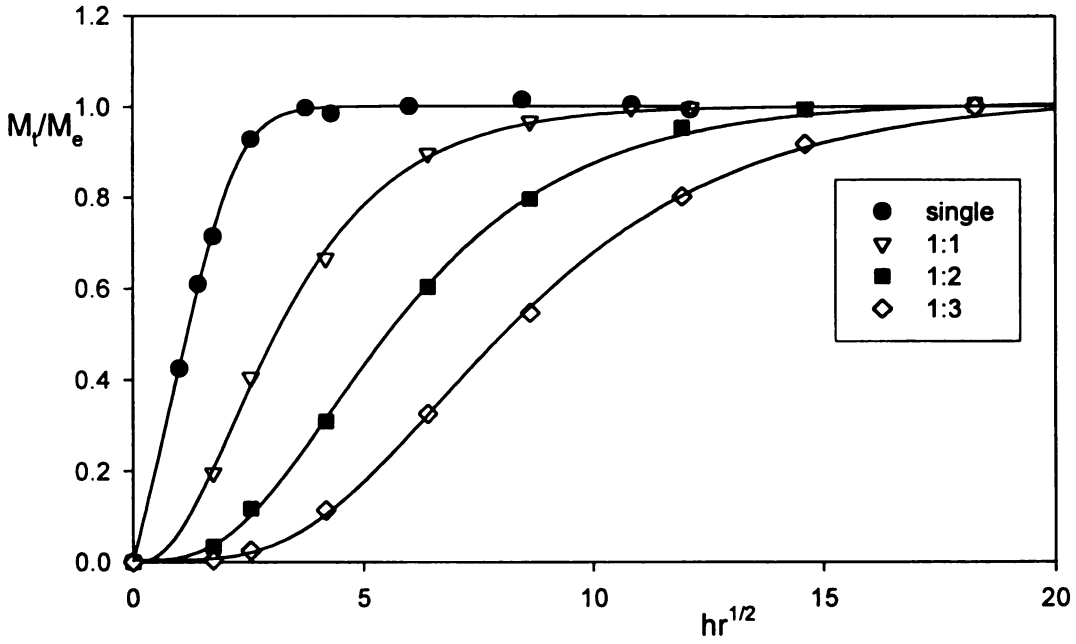


Figure 5.9 Effect of thickness of functional barrier on migration rate of BHT to 100% ethanol at 31°C

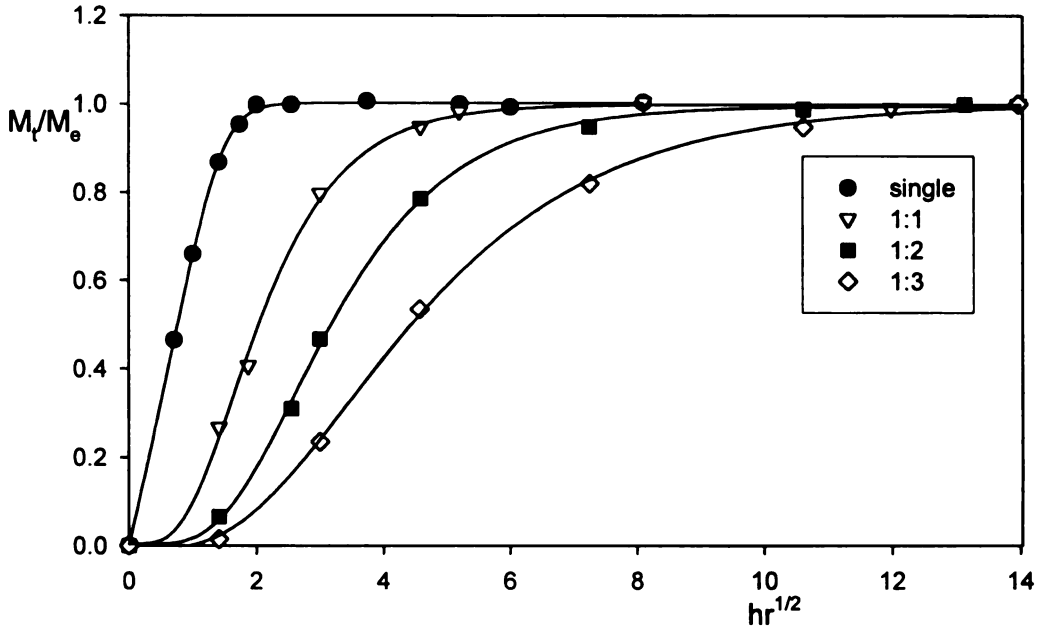


Figure 5.10 Effect of thickness of functional barrier on migration rate of BHT to 100% ethanol at 40°C

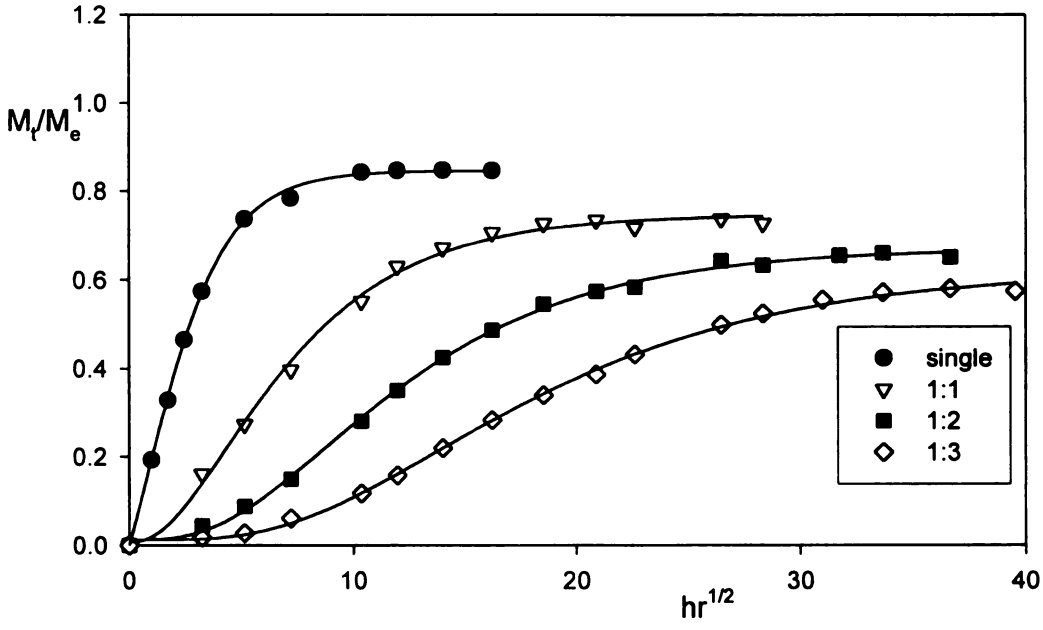


Figure 5.11 Effect of thickness of functional barrier on migration rate of BHT to 50% ethanol at 23°C

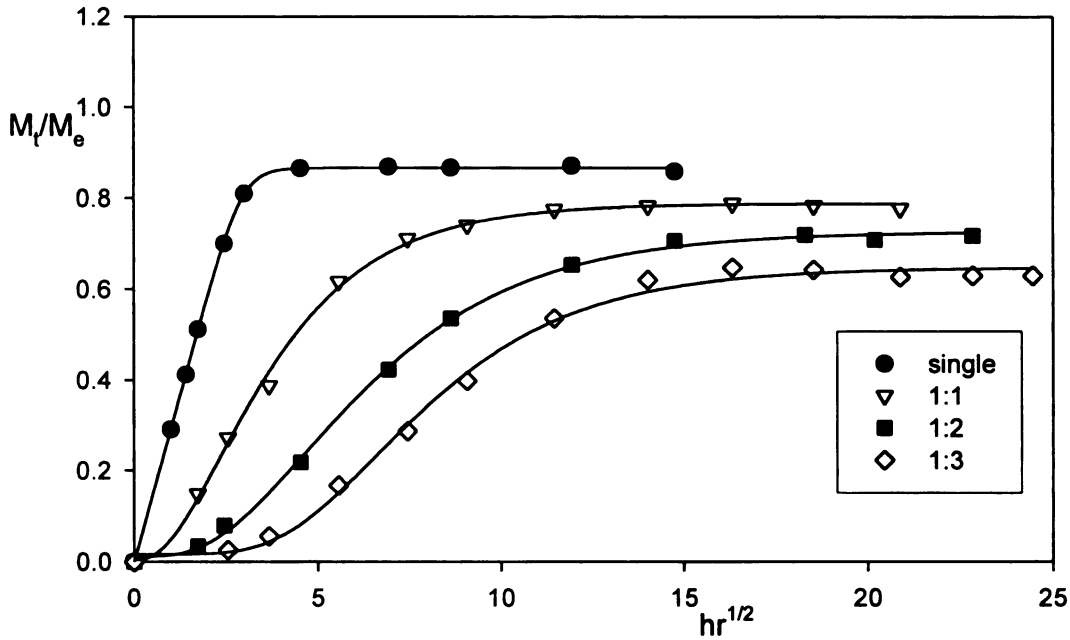


Figure 5.12 Effect of thickness of functional barrier on migration rate of BHT to 50% ethanol at 31°C

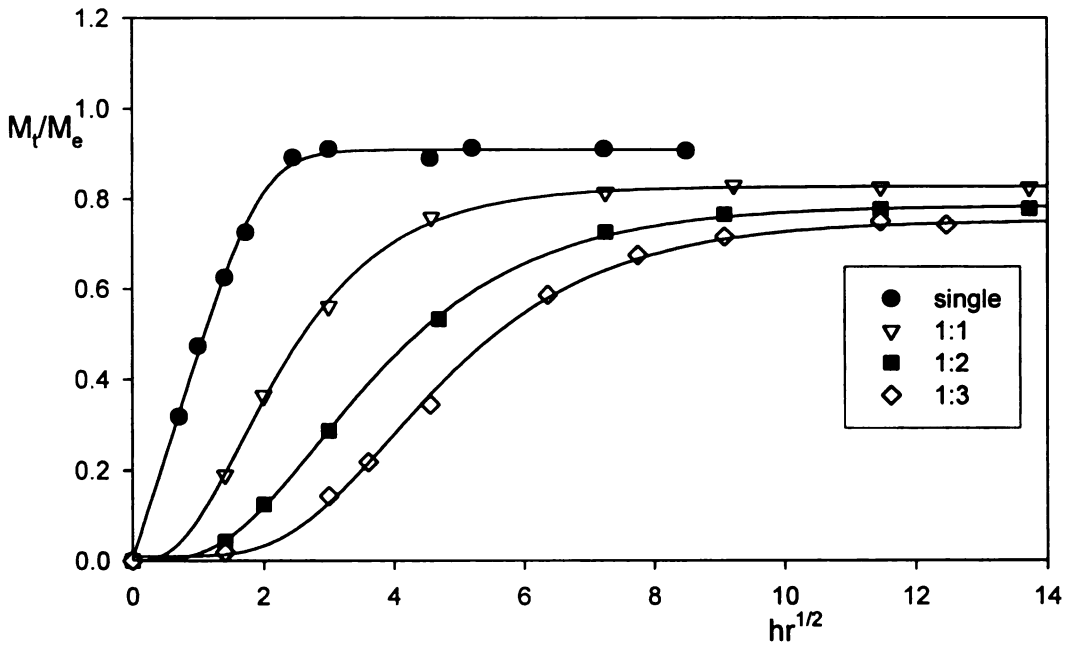


Figure 5.13 Effect of thickness of functional barrier on migration rate of BHT to 50% ethanol at 40°C

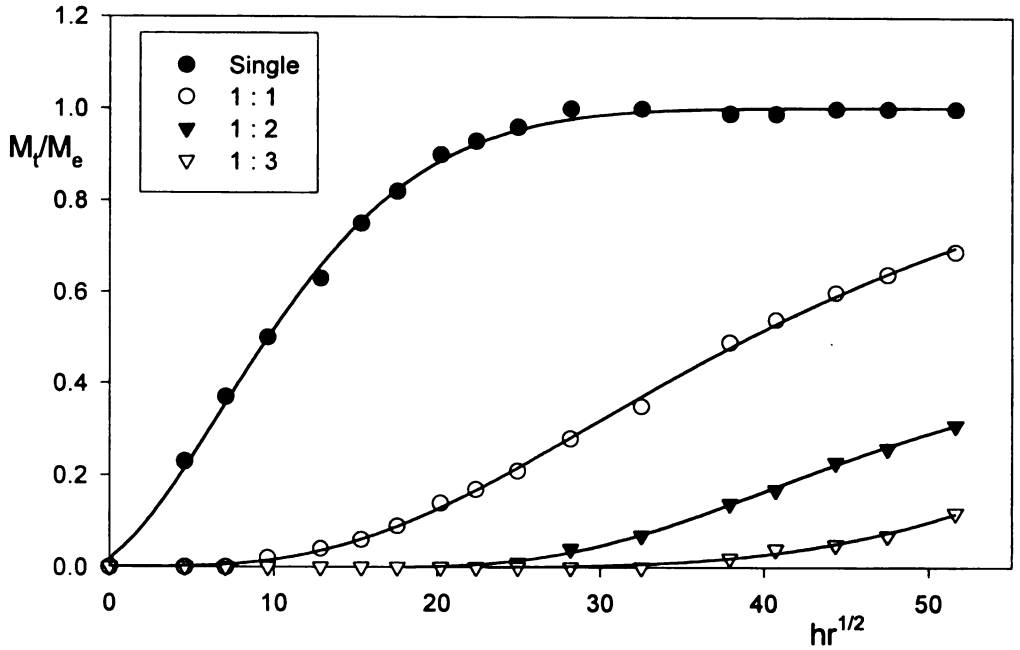


Figure 5.14 Effect of thickness of functional barrier on migration rate of Irganox™1076 to 100% ethanol at 23°C

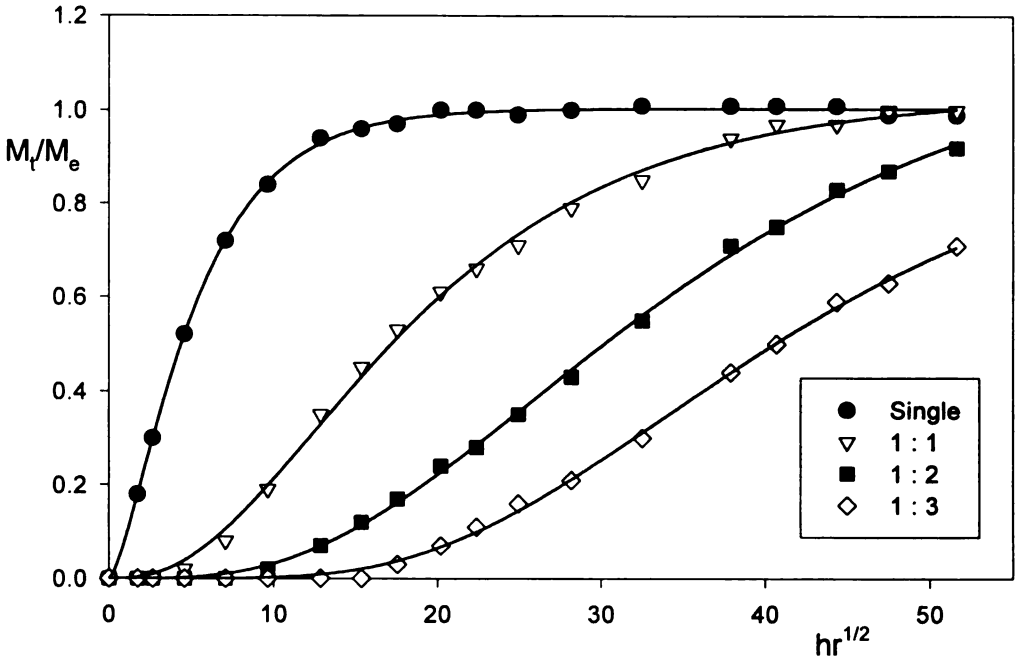


Figure 5.15 Effect of thickness of functional barrier on migration rate of Irganox™1076 to 100% ethanol at 31°C

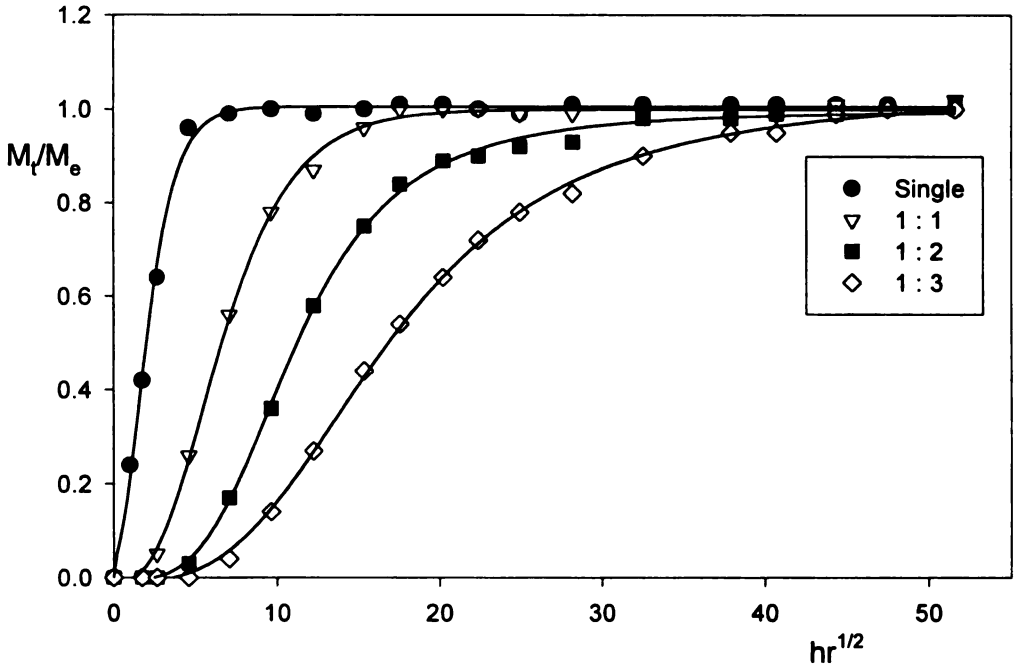


Figure 5.16 Effect of thickness of functional barrier on migration rate of Irganox™1076 to 100% ethanol at 40°C

Table 5.12 Storage time for the migration process to reach equilibrium for BHT

(Unit: hours)

Temperature, °C	Effective Thickness* Ratio			
	1 : 0	1 : 1	1 : 2	1 : 3
23	50	280	650	1200
31	15	75	180	320
40	6	30	70	120

* For the double sided extraction method, the effective thickness is one half of the core layer and the full thickness of one outer layer.

5.4. Comparison of Computer Model to Existing Analytical Models

Theoretical models have been developed for the prediction of the migration of a contaminant from the core layer of a multilayer polymer system into a food simulant. Equation 2.9 is the model proposed by Begley and Hollifield (1993) and Equation 2.10 is an expression developed by Laoubi and Vergnaud (1995). These two models were compared with the computer simulation model developed in this study.

For the case of the same effective thickness of the core and outer layer at 40°C, the experimentally obtained and theoretically determined parameters (Table 5.13) were incorporated into Eqation 2.9, Equation 2.10 and the computer model, and the results of the calculations and the actual experimental results were plotted in the same graph (Figure 5.17).

The Begley and Hollifield model greatly over-estimated the migration amount due to over-simplification of the parameters and assumptions. Estimation by the computer model correlated very well with not only the actual experimental results but also with the calculation by the Laoubi & Vergnaud model, which helps to confirm the validity of the finite element-based computer model.

- Begley & Hollifield Model (Equation 2.9)

$$\frac{M_{F,t}}{M_{F,e}} = \frac{Dt}{(L-R)^2} - \frac{1}{6} - \frac{2}{\pi^2}\sum_{1}^{\infty}\frac{(-1)^n}{n^2}\exp\left(\frac{n^2\pi^2 Dt}{(L-R)^2}\right)$$

- Laoubi & Vergnaud Model (Equation 2.10)

$$\frac{M_{F,t}}{M_{F,e}} = 1 - \frac{8}{\pi^2}\frac{L}{R}\sum_{1}^{\infty}\frac{(-1)^n}{(n+1)^2}\sin\left[\frac{(2n+1)\pi R}{2L}\right]\exp\left[-\frac{(2n+1)^2\pi^2 Dt}{4L^4}\right]$$

Table 5.13 Parameters for estimation of migration

Model	Effective thickness, mil (Core : Outer)	Diffusion Coefficient, D, Cm^2/sec	
		Core layer	Outer layer
Begley & Hollifield	0.6 : 0.6	$1.98 \times 10^{-10*}$	$1.98 \times 10^{-10*}$
Laoubi & Vergnaud	0.6 : 0.6	$1.98 \times 10^{-10*}$	$1.98 \times 10^{-10*}$
Computer model	0.6 : 0.6	2.20×10^{-10}	1.76×10^{-10}

* Weighted average of diffusion coefficients of core and outer layer

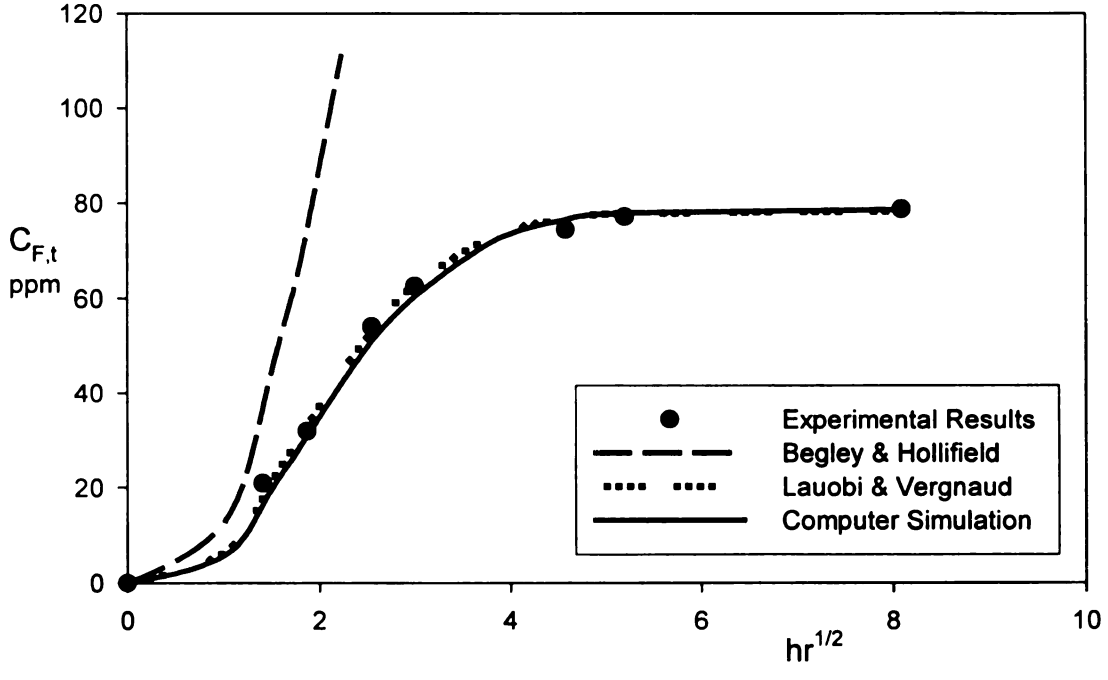


Figure 5.17 Comparison of migration models

5.5. Applications of the Model/Computer Program

5.5.1. Single Layer

A model of BHT migration through the core single layer to 100% ethanol (without partitioning) was designed as shown in Figure 5.18 for the FEM analysis. Figure 5.19 compares the output from the computer program with the actual experimental results (dots on the graph) for the test temperatures, which shows good agreement with actual experimental results.

5.5.2. Structure with Functional Barrier Layer

Migration from a multilayer system with a functional barrier can be modeled by simply assigning appropriate nodal values for the particular nodes. Figure 5.20 shows the system design for contaminant migration through the core and outer layers when the effective thickness ratio is 1:1. Figures 5.21 to 5.23 for BHT and Figures 5.24 to 5.26 for Irganox™1076 migration show outputs from the computer program, compared with the actual experimental results (dots in the graph) for each temperature, which demonstrate fairly good agreement with the experimental results. In this study, the diffusion coefficients in the outer layer were estimated with empirical equations; however, application of the model with actually measured diffusion coefficients must be a critical factor.

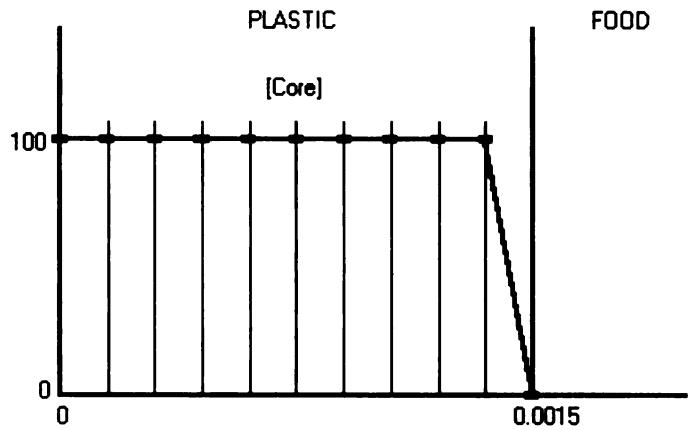


Figure 5.18 System design for single layer migration

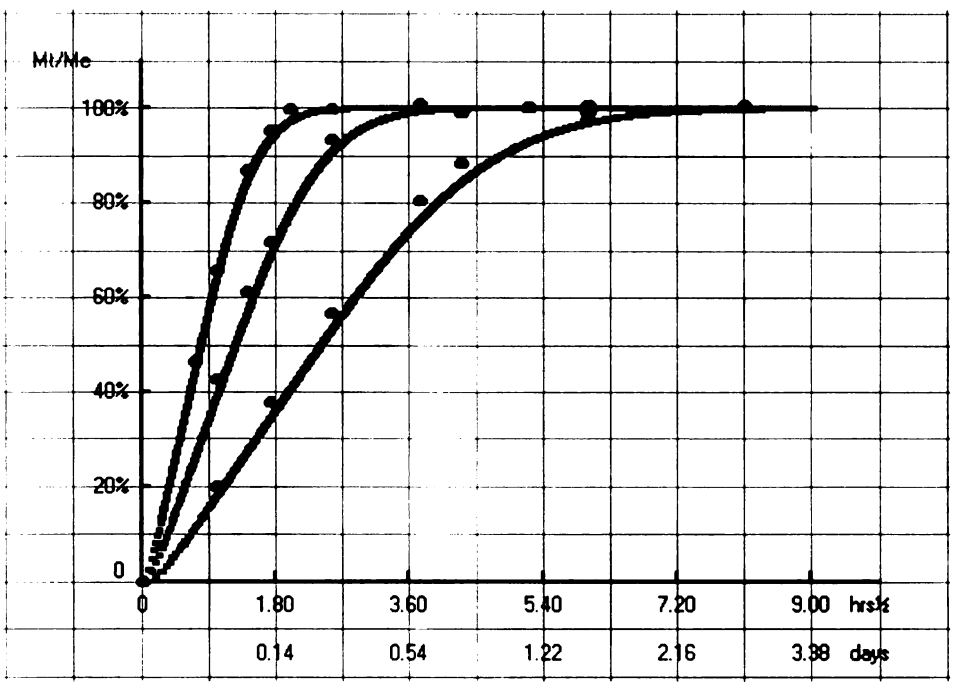


Figure 5.19 Output for BHT migration through the core single layer to 100% ethanol at 23, 31, and 40°C (from right to left)

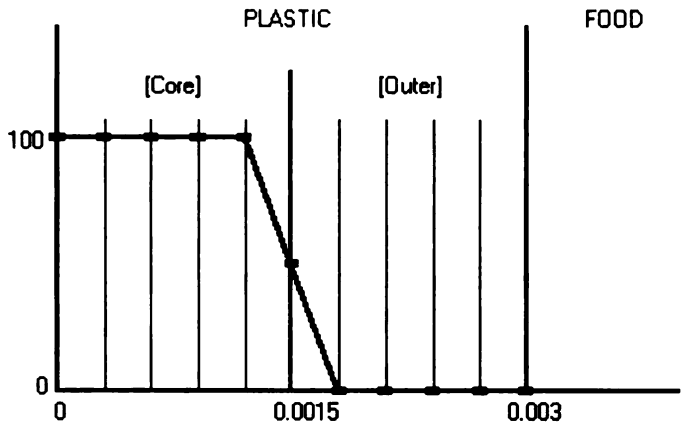


Figure 5.20 System design for multilayer migration

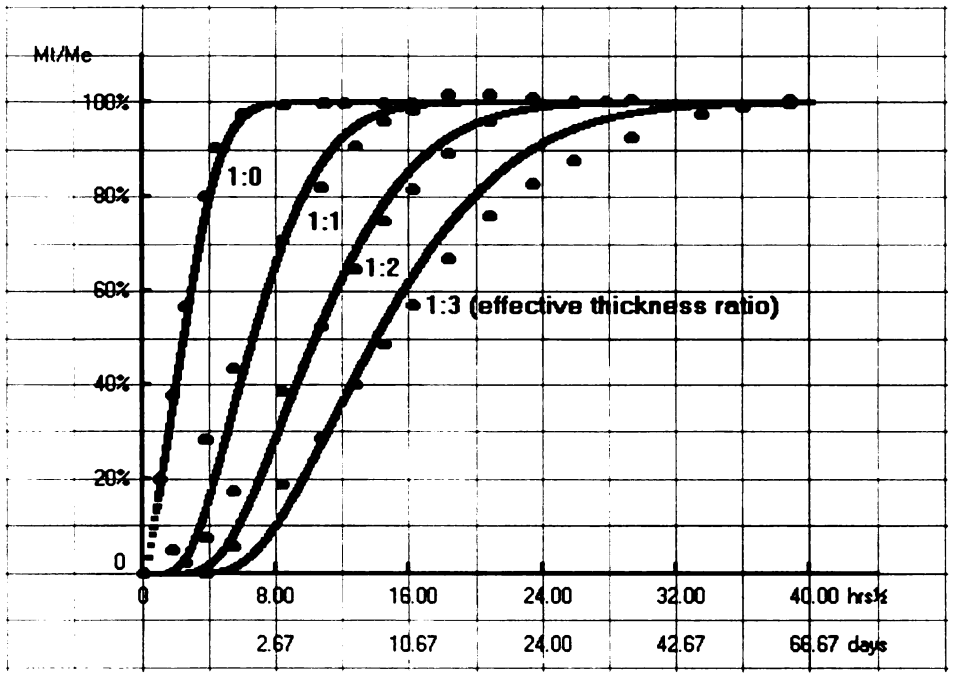


Figure 5.21 Output for BHT migration through the core and outer layers with varying thickness to ethanol at 23°C

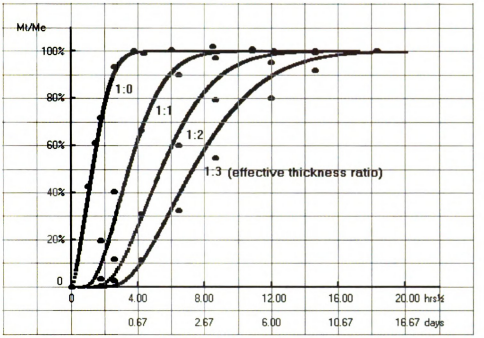


Figure 5.22 Output for BHT migration through the core and outer layers with varying thickness to ethanol at 31°C

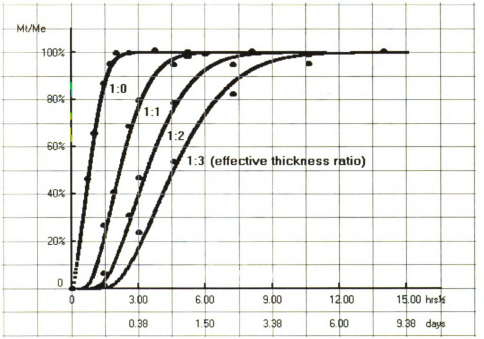


Figure 5.23 Output for BHT migration through the core and outer layers with varying thickness to ethanol at 40°C

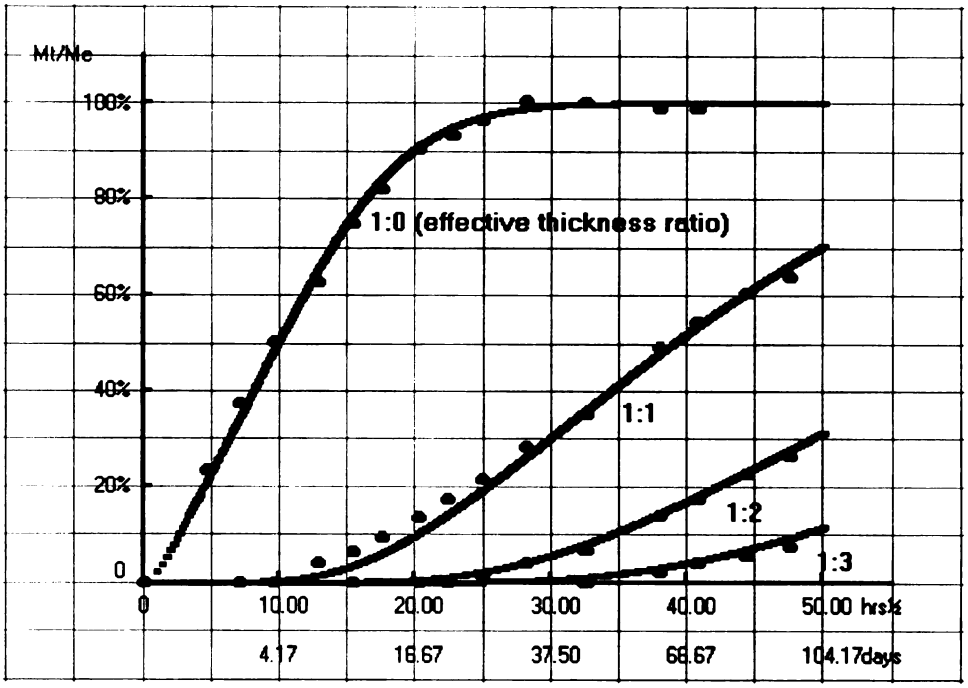


Figure 5.24 Output for Irganox™1076 migration through the core and outer layers with varying thickness to ethanol at 23°C

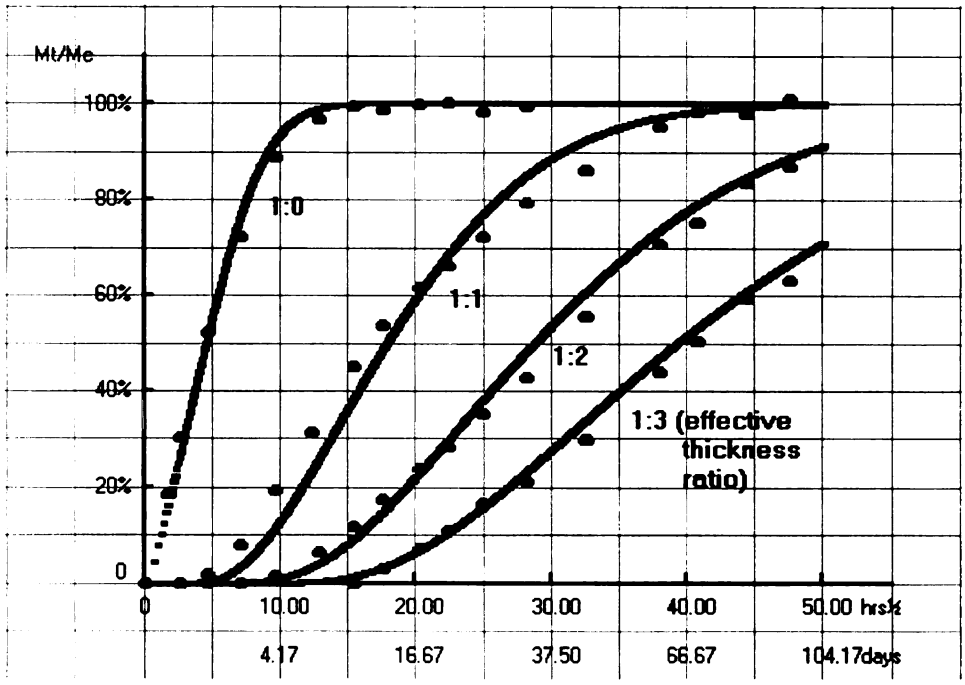


Figure 5.25 Output for Irganox™1076 migration through the core and outer layers with varying thickness to ethanol at 31°C

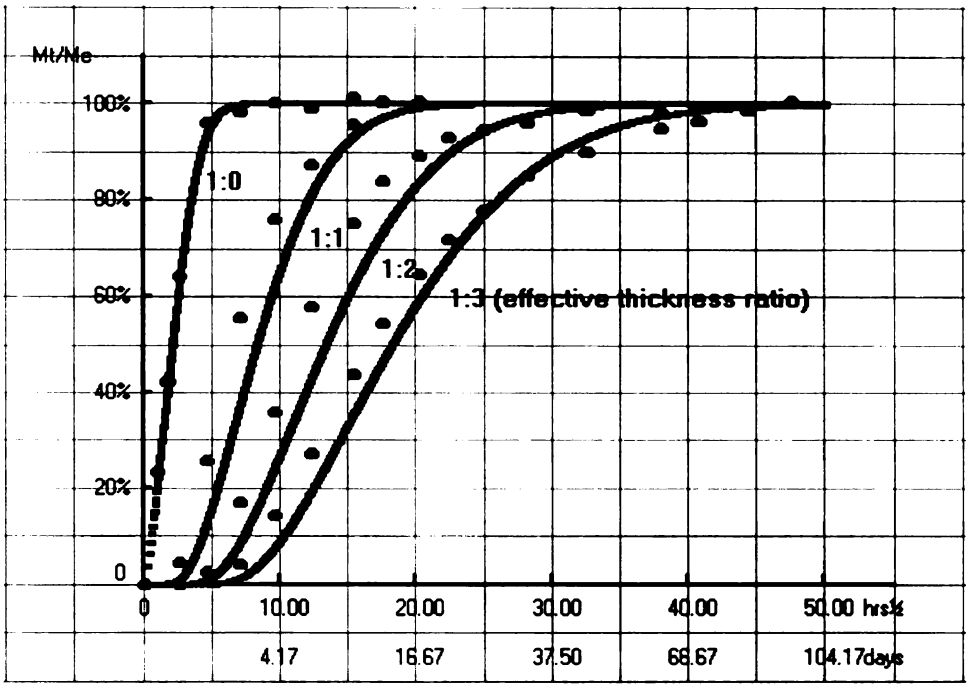


Figure 5.26 Output for Irganox™1076 migration through the core and outer layers with varying thickness to ethanol at 40°C

5.5.3. Migration with Partitioning

Partitioning of the migrant between the food/liquid and polymer often plays a key role in affecting the rate of migration as well as in determining the ultimate total quantity of migrant lost from the polymer or gained by the food.

A system of migration with partitioning can be modeled as shown in either Figure 5.18 or Figure 5.20, depending on the presence of a functional barrier. Figure 5.27 shows an output from the computer program for BHT migration through the polymer with functional barrier (effective thickness ratio = 1:2) to the food simulants at 31 ^{o}C. The upper curve represents the migration simulation for the system without partitioning (migration to 100% ethanol) and the lower curve shows the simulation for the partitioning situation (migration to 50% aqueous ethanol), compared with the actual experimental results (dots on the graph).

Figures 5.28 to 5.30 show the outputs from the computer program for BHT migration through the polymer with functional barriers of varying thickness to the 50% aqueous ethanol. Compared with the actual experimental results (dots on the graph), in general, the computer model shows somewhat overestimation of migration. It is assumed in this study that a partition coefficient is constant throughout the migration process; however, polymers are heterogeneous materials, with amorphous and crystalline domains, so that the physical meaning of a constant partition coefficient may be questioned.

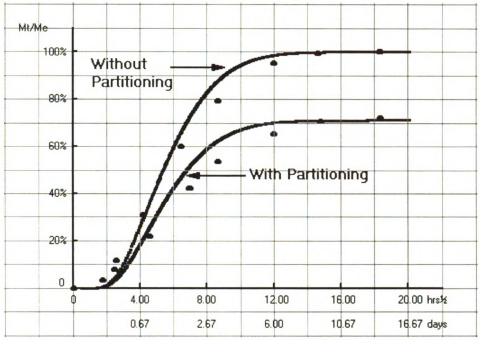


Figure 5.27 Output for BHT migration through the polymer with functional barrier to ethanol at 31°C with/without partitioning

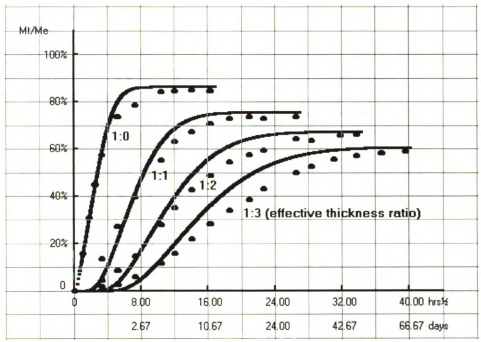


Figure 5.28 Output for BHT migration through the core and outer layers with varying thickness to 50% aqueous ethanol at 23°C

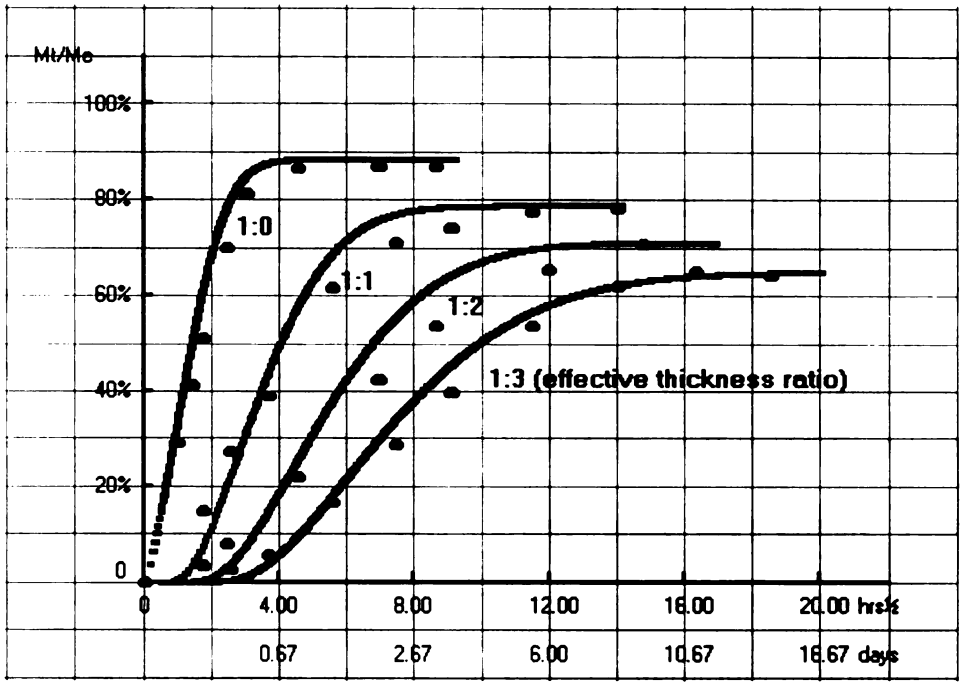


Figure 5.29 Output for BHT migration through the core and outer layers with varying thickness to 50% aqueous ethanol at 31°C

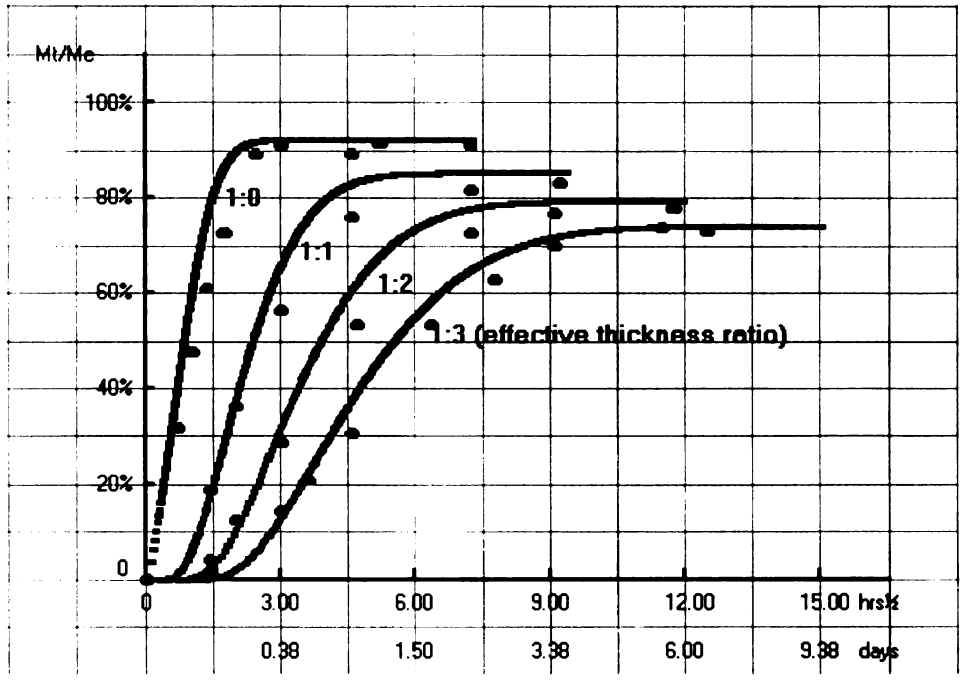


Figure 5.30 Output for BHT migration through the core and outer layers with varying thickness to 50% aqueous ethanol at 40°C

Hamdani et al (1997) pointed out that a partition coefficient might decrease as migration proceeds due to heterogeneous distribution of the potential migrants in the polymer. Such Langmuir sorption behavior (Adamson and Gast, 1997) has been clearly demonstrated in the case of vinyl chloride monomer in polyvinyl chloride (Kinigakis et al, 1987).

However, considering that calculation of answers using analytical solutions tends to be very complicated, the computer program can be an efficient practical tool for evaluating consumers' safety, since the worst case scenario plays a major role in the modeling of contaminant migration.

5.5.4 Migration from Multilayer Structures with Very Different Diffusion Coefficients

One of the unique advantages of the finite element method is the ability to assign a different diffusion coefficient, D, independently for each element and there is practically no limitation on values of the diffusion coefficients. Therefore, the computer model developed with the FEM can be used to estimate the migration from multilayer structures with very different diffusion coefficients.

For the case of the same effective thickness of the core and outer layer (0.6 mil), various diffusion coefficients for the outer layer were assumed (Table 5.14) and incorporated into the computer model, and the results of simulation by the computer model were plotted in the same graph (Figure

5.31). The effect of decreasing the permeability of the functional barrier layer can be clearly seen.

Table 5.14 Various diffusion coefficients for estimation of migration

Layer	Thickness, cm	Diffusion Coefficient of Contaminant, cm^2/sec
Core layer	0.0015	2.20×10^{-10}
Virgin layer	0.0015	
Case (a)		2.20×10^{-10}
Case (b)		2.20×10^{-11}
Case (c)		2.20×10^{-12}
Case (d)		2.20×10^{-13}

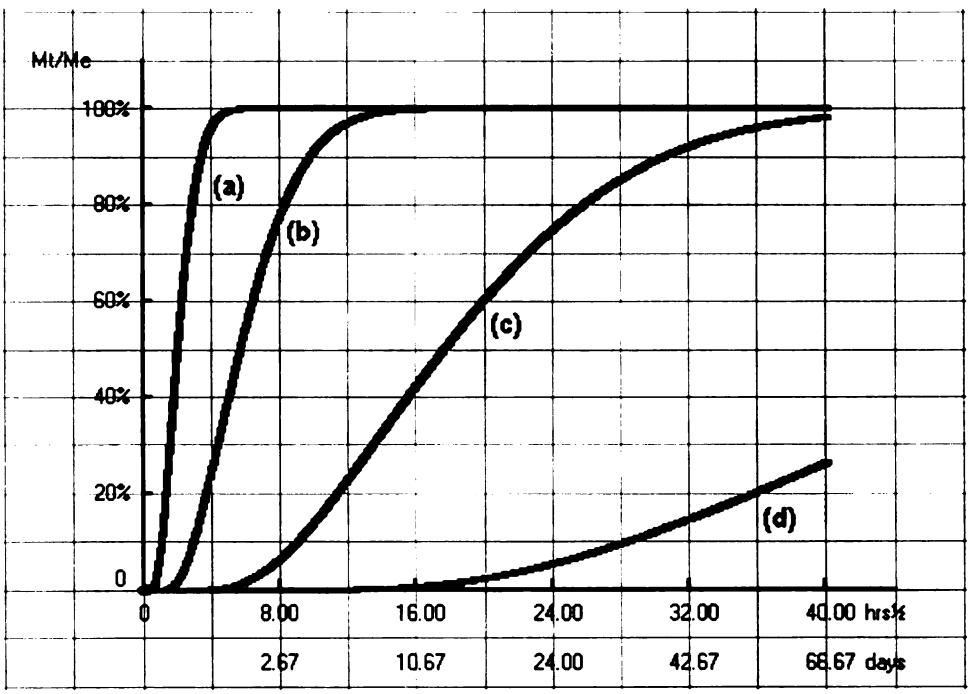


Figure 5.31 Output for migration with various diffusion coefficients of contaminant in the outer layer

5.5.5. Diffusion During Extrusion and Storage of Polymers

Since the multilayer functional barrier structures are mostly manufactured by co-extrusion processes applying high temperatures far above the melting points of the polymers, significant diffusion just after extrusion may be expected between the core and virgin functional barrier. Franz et al (1977) developed a physico-mathematical model which took into account an already existing contaminant in the functional barrier layer and attempted to verify the model. The same effect can occur if the multilayer polymer structures are stored for long periods of time before the packaging application. Piringer et al (1998) presented a general mathematical model which permitted the amount of migration through a functional barrier to be predicted even when the barrier layer becomes contaminated during polymer processing and/or storage.

One of the unique advantages of the finite element method is the ability to assign a value independently for each node; for the diffusion study, the concentration value of the contaminant at a certain point in the film can be assigned arbitrarily. Figure 5.32 (a) represents an ideal concentration distribution (no diffusion) in the multilayer extruded film. Figure 5.32 (b) shows a more realistic situation for the concentration distribution within the polymer structure, with an assumption of about 40% transfer by diffusion of the contaminant into the outer layer during extrusion and storage. For both cases, the shaded areas representing the total amount of contaminant in the polymer are identical. Figure 5.33 shows an output from the computer program for

BHT migration through a polymer with functional barrier (effective thickness ratio = 1:2) to ethanol at 31°C. The left hand curve represents the migration simulation for the system with diffusion during extrusion and storage and the right hand curve shows the simulation for the ideal case. If the amount of contaminant diffusion during extrusion and storage is known precisely, the migration rate of the contaminant to the food can be accurately modeled.

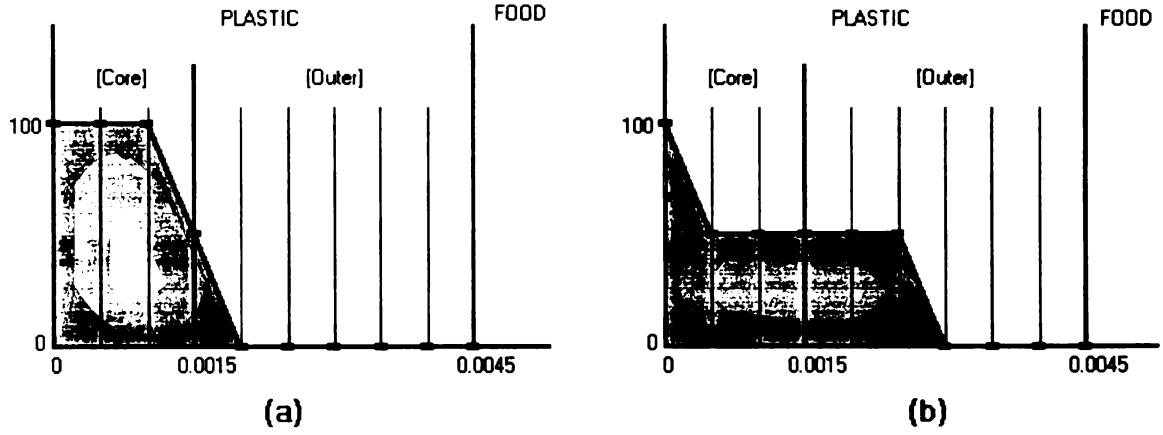


Figure 5.32 System design for migration after diffusion during extrusion/ storage

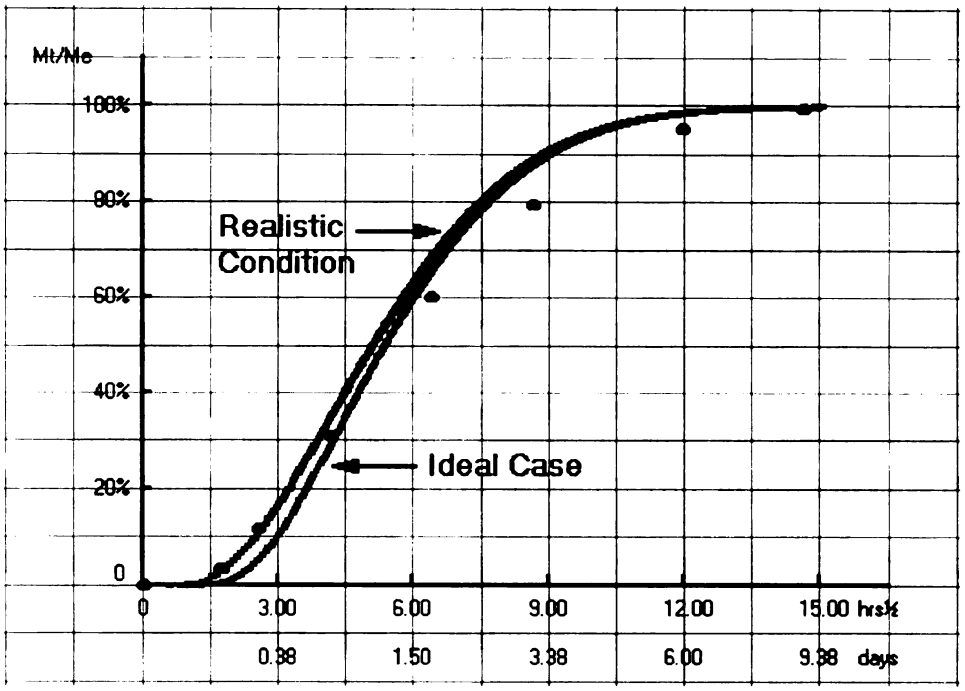


Figure 5.33 Output for BHT migration to ethanol at 31°C - realistic condition

5.5.6. Migration with Back-Diffusion to Polymer by Food/Liquid

The packaged food/liquid product or its aggressive components can penetrate polymer packaging materials, which is often called back-diffusion or swelling. The back-diffused food or liquid component may play a role as an additive in the polymer; as a result, the diffusion coefficient of the migrant increases in the swollen region. Rudolph (1979) has shown that if the liquid diffuses into the polymer while the migrant diffuses out, it would still be expected that migration would correlate with the square root of time, but the effective diffusion coefficient would be different. Gandek et al. (1989) pointed out the importance of product back-diffusion in the migration process. Figge (1996) defined this phase boundary as the "swelling layer", and suggested

that the diffusion coefficient of the migrant in the swelling layer must be distinctly greater than the diffusion coefficient in the unchanged polymer region.

This complicated migration situation can be modeled easily with the worst case scenario concept and the unique characteristics of the finite element method. With the known values of the maximum swelling depth (the worst case) and the diffusion coefficient of the migrant in the swelling region, the system can be designed by dividing the thickness appropriately and by selecting the diffusion coefficient correctly.

Figure 5.34 shows an example of the system design for BHT migration through the core and outer layers with swelling (shaded area) at 40°C, and the assumed migration parameters are summarized in Table 5.15. Figure 5.35 shows output from the computer program compared with the actual experimental results (dots on the graph).

Table 5.15 Parameters for estimation of migration with back-diffusion

Layer	Thickness, cm	Diffusion Coefficient of BHT, cm^2/sec
Core layer	0.0015	2.20×10^{-10}
Virgin layer	0.0045	
- Unchanged	0.0040	1.76×10^{-10}
- Swollen	0.0005*	1.76×10^{-8}*

* Assumed values

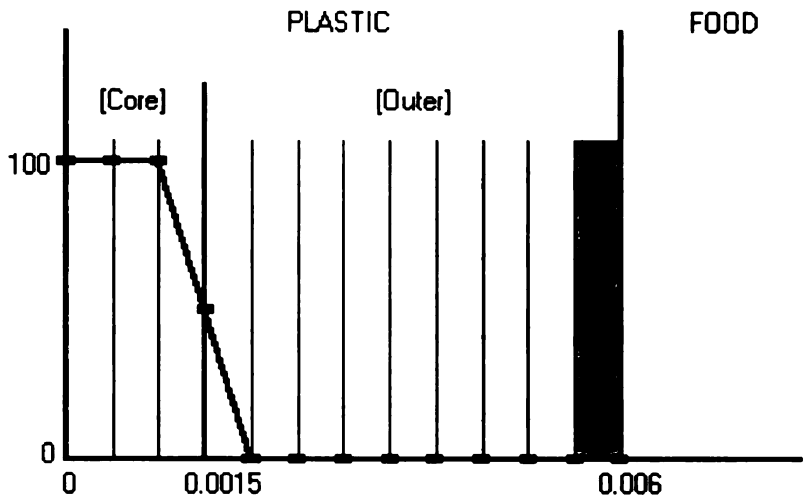


Figure 5.34 System design for migration with swelling; the shaded area represents the maximum swollen region

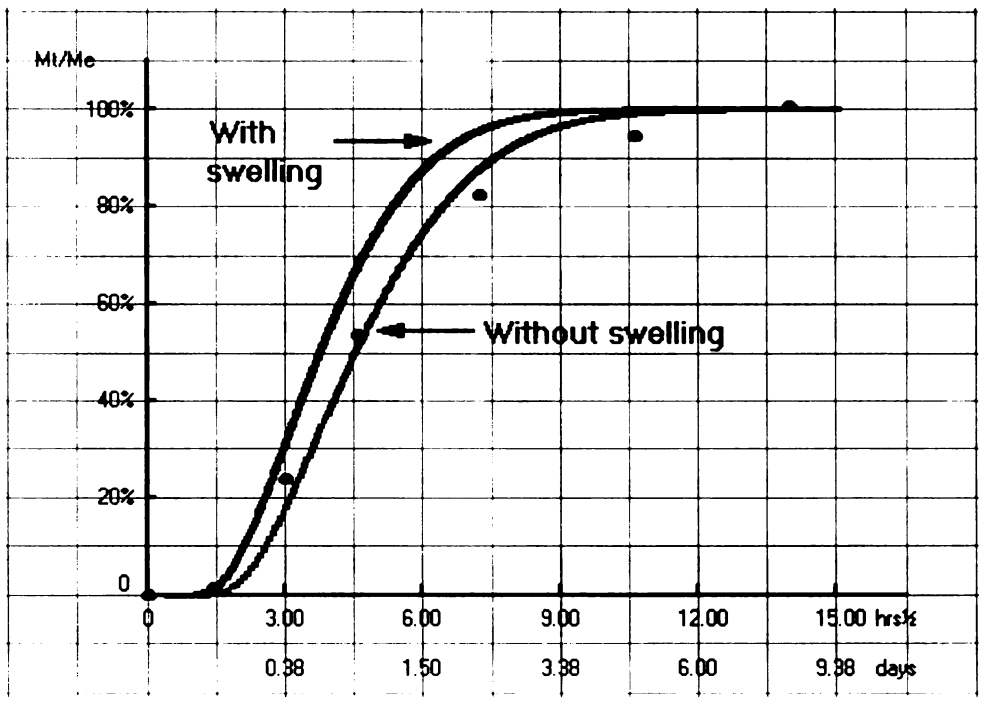


Figure 5.35 Output for BHT migration with assumption of swelling through the core and outer layers to ethanol at 40°C

5.5.7 Considerations for Practical Applications of the Computer Simulation Model

The basic concept of the modeling in this study is diffusion of the contaminant symmetrically from the core layer through the barrier layers and into the food, with no resistance to transfer at the virgin layer/food interface. Therefore, migration was modeled as one-dimensional and the polymer represented by half the structure: from the center of the core layer to the outside of the virgin layer.

However, the 'real world' situation is usually somewhat different from the basic assumptions of the model. One virgin functional barrier (inside) will directly contact the liquid food, meanwhile the other (outside) will in practice contact the air during physical distribution and storage. If the contaminant is extremely volatile, then it may show a tendency to evaporate into the air rather than to migrate to the food; otherwise it may preferably migrate to the food. In the latter case, the other half of the core layer may play a role as a 'reservoir' of the contaminant in the 'real world' situation, and eventually most of the contaminant may migrate to the food.

Considering that the worst case scenario plays a major role in the modeling of contaminant migration to ensure consumers' safety, the whole thickness of the core layer can be used with one virgin functional barrier layer for calculation of one directional migration.

Figure 5.36 shows a comparison of the results of simulation for BHT migration at 40°C, using the whole thickness (1.2 mil) as a core layer and

using one half of the thickness (0.6 mil) as a core layer, when the thickness of the outer layer was kept at 0.6 mil.

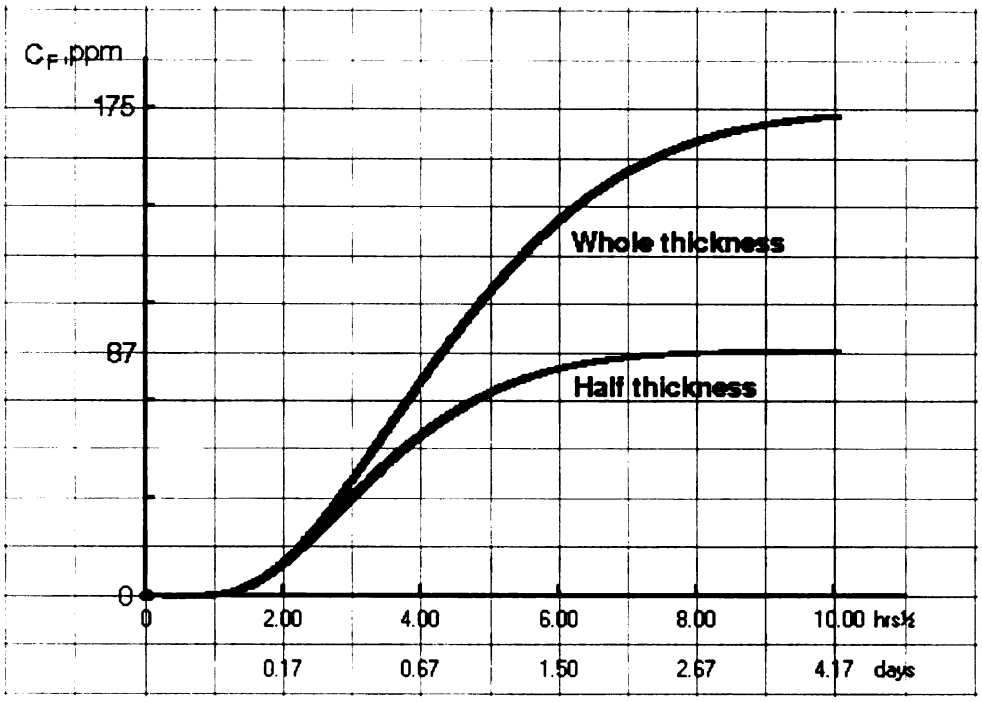


Figure 5.36 Comparison of simulation for BHT migration at 40°C between whole thickness and one half of the thickness as a core layer

6. CONCLUSIONS

6.1 Summary

Migration can be accurately predicted and modeled if all the migration controlling parameters are known. The FDA has already accepted the use of model simulants and the Begley and Hollifield diffusion model as the basis for approval of functional barrier structures (Begley & Hollifield, 1993). However, more accurate models could make it possible to utilize plastics with less barrier ability as functional barriers for food packaging applications.

In this study, a computer model which accurately reflects the physics of the migration process from the multilayer structure to the liquid food simulant was developed using a numerical treatment, the finite element method. The accuracy of the model in predicting migration was successfully demonstrated by comparing the simulated results to experimental data. It was also demonstrated that the computer model using the finite element method can be a versatile tool to estimate the migration of the contaminant for many practical problems, considering that a reasonably worst case scenario is always acceptable.

Experimentally, this study examines the effectiveness of a virgin HDPE layer as a "functional barrier" in reducing migration of contaminants from a recycled core layer into a food. The amount of migration of the contaminant

simulant from the core layer to the liquid food simulants was determined as a function of the thickness of the outer layer at different temperatures.

Results indicated that the thicker "functional barrier" apparently provided better protection of the food from contamination with substances in the recycled layer; however, economical and technical feasibility of the thicker material should be justified before designing thick barrier layers. The efficiency of a "functional barrier" may be increased dramatically by applying better barrier material, including OPP, PET and multi-resin systems as functional barriers, since the diffusion coefficient is more influential for the migration process than thickness.

The major emphasis of this research has been to develop a computer program which can mathematically predict the migration process for multilayer structures. This computer program, developed as a total solution package for migration problems, can be applied not only to multilayer structures made with the same plastics but also with different plastics. Any kind of laminated or co-extruded structure can be modeled if the related parameters are known. Furthermore, diffusion of substances inside the polymer during the extrusion process and storage of film stocks can be incorporated into the model. Finally the model can be a useful tool in determining what type of barrier will be effective for certain multilayer structures and/or packaging systems.

6.2 Recommended Future Work

1) Due to the difficulty in preparing the samples, the diffusion coefficients were estimated with empirical equations for some cases. Application of the model with measured diffusion coefficients is recommended.
2) The model developed in this study can be applied to laminated structures, taking into consideration that an adhesive layer of the laminated polymer is extremely thin; the model should be effective with that system.
3) A standard method to determine diffusion coefficients should be studied; some authors recommended the film-to-film method, in which an efficient or apparent diffusion coefficient is measured for a specific system. The diffusion coefficient should be an intrinsic property of a substance in a polymer and only dependent on the temperature.
4) Since we were unable to acquire samples containing the contaminant simulants recommended by FDA, in this study phenolic antioxidants were selected as the contaminant simulants. To get acceptance of the model by FDA, migration experiments with the contaminant simulants recommended by FDA are desirable.
5) The computer program developed in this study may be the first effort to apply the finite element method to multi-layer packaging systems; the author will be grateful if there are modifications of this program in the future.

APPENDICES

APPENDIX A

STANDARD CALIBRATION CURVE

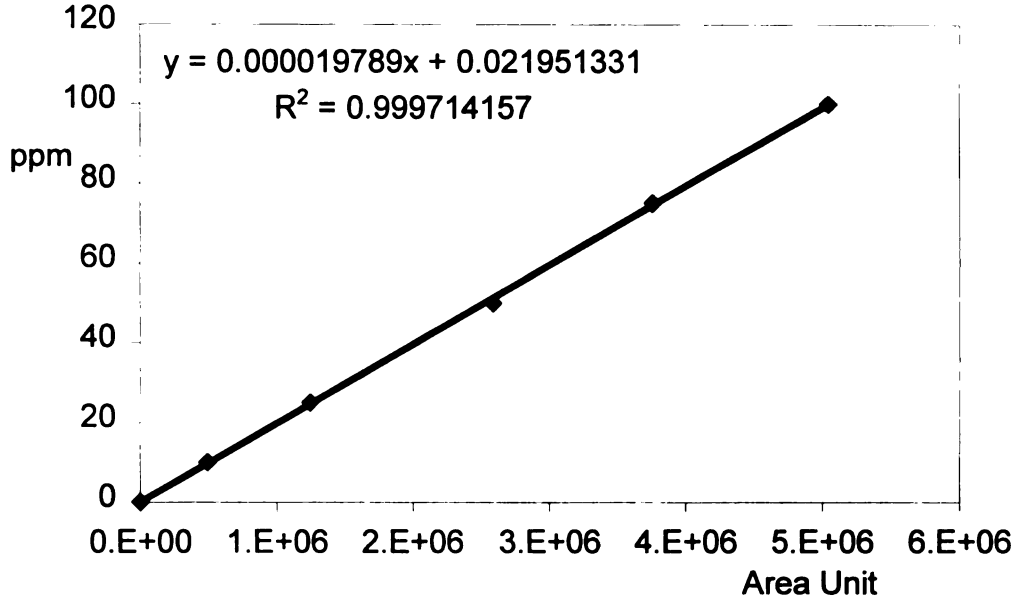


Figure 6.1 Calibration curve for BHT in acetonitrile

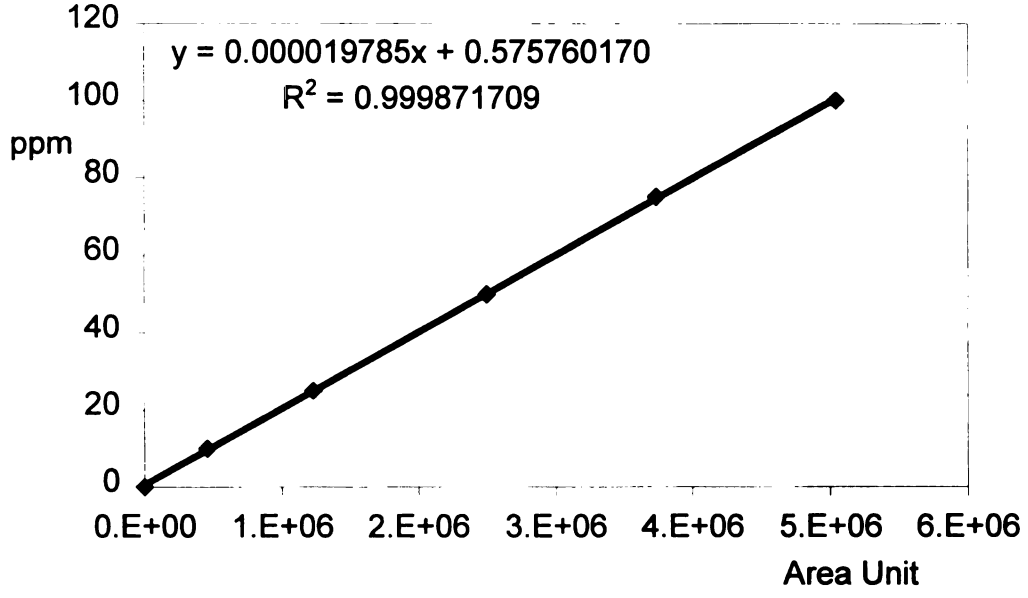


Figure 6.2 Calibration curve for BHT in 100% ethanol

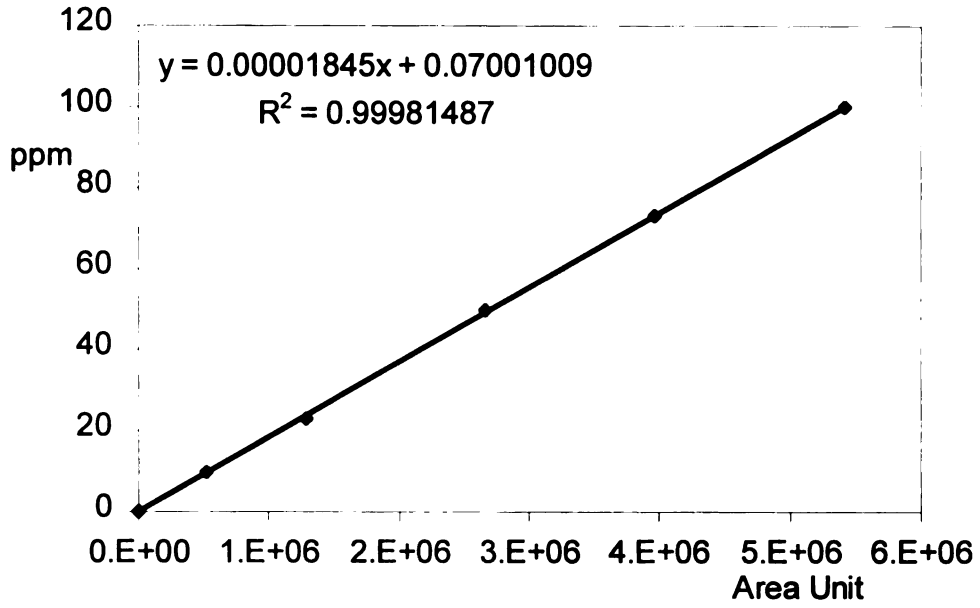


Figure 6.3 Calibration curve for BHT in 50% ethanol

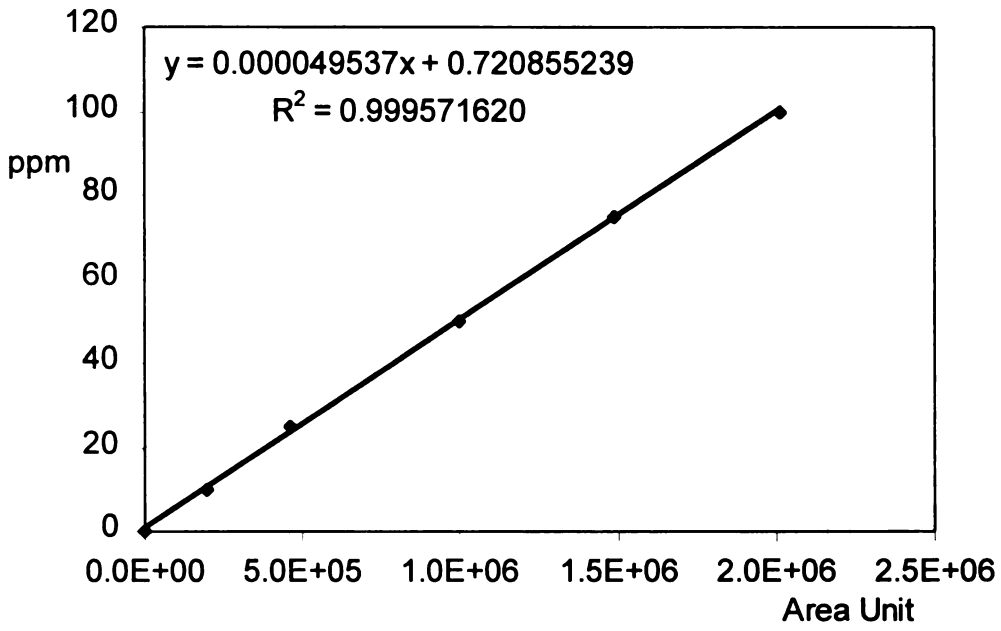


Figure 6.4 Calibration curve for Irganox™1076 in methylene chloride

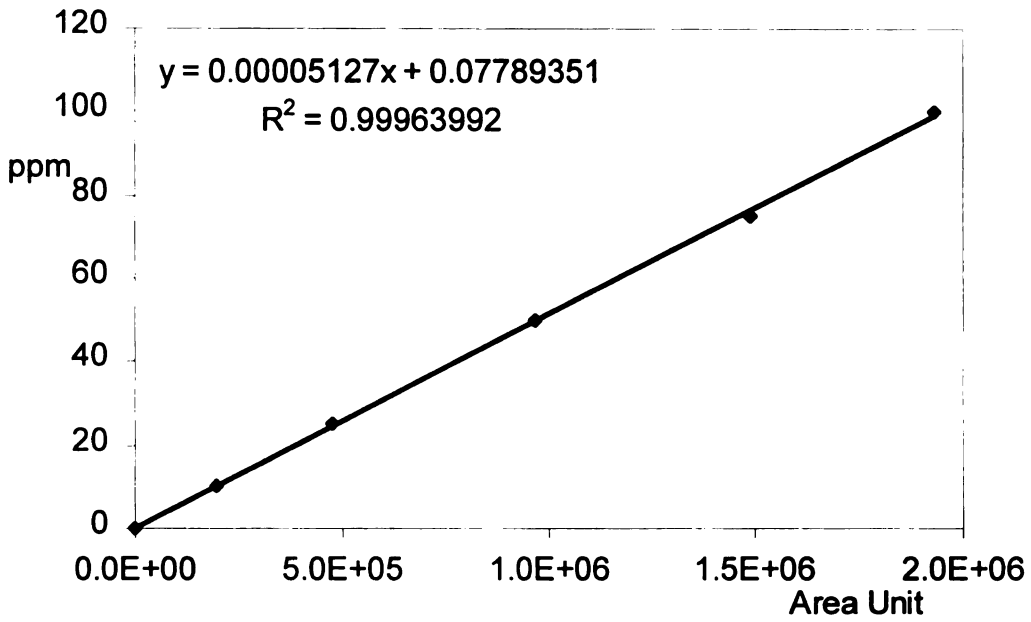


Figure 6.5 Calibration curve for Irganox™1076 in 100% ethanol

APPENDIX B

COMPUTER PROGRAM CODE

Form

"frmMain" Code

```
Private Declare Function OSWinHelp% Lib "user32" Alias "WinHelpA" (ByVal hwnd&, ByVal
HelpFile$, ByVal wCommand%, dwData As Any)

Private Sub MDIForm_Load()
   Me.Left = GetSetting(App.Title, "Settings", "MainLeft", 1000)
   Me.Top = GetSetting(App.Title, "Settings", "MainTop", 1000)
   Me.Width = GetSetting(App.Title, "Settings", "MainWidth", 6500)
   Me.Height = GetSetting(App.Title, "Settings", "MainHeight", 6500)
   frminput5.Show Modal
End Sub

Private Sub MDIForm_Unload(Cancel As Integer)
   If Me.WindowState <> vbMinimized Then
      SaveSetting App.Title, "Settings", "MainLeft", Me.Left
      SaveSetting App.Title, "Settings", "MainTop", Me.Top
      SaveSetting App.Title, "Settings", "MainWidth", Me.Width
      SaveSetting App.Title, "Settings", "MainHeight", Me.Height
   End If
End Sub

Private Sub mnuFileOpen_Click()
Const conbtns = vbYesNoCancel + vbExclamation + vbDefaultButton3 + vbApplicationModal
Dim udtitem As itemstruc
Dim Filename As String
   On Error GoTo OpenErrHandler
   dlgCommonDialog.CancelError = True
   dlgCommonDialog.Flags = cdlOFNFileMustExist + cdlOFNOverwritePrompt
   dlgCommonDialog.Filter = "Names data files(*.mig)|*.mig|All Files (*.*)|*.*"
   dlgCommonDialog.Filename = ""
   dlgCommonDialog.ShowOpen
   Open dlgCommonDialog.Filename For Random As #1 Len = Len(udtitem)
   Get #1, 97, udtitem
      frminput5.NumNodesi.Text = udtitem.NumNodesi:  frminput5.NumElei.Text = udtitem.NumElei
      frminput5.NumMatlSetsi.Text = udtitem.NumMatlSetsi
      frminput5.NumKwnPhii.Text = udtitem.NumKwnPhii
      frminput5.txtCorelayer.Text = udtitem.txtCorelayer: frminput5.txtOutlayer.Text = udtitem.txtOutlayer
      frminput5.txtExtralayer.Text = udtitem.txtExtralayer
      frminput5.txtTotalThick.Text = udtitem.txtTotalThick
      frminput5.txtTotalMil.Text = udtitem.txtTotalMil
      For I = 1 To 3
          frminput5.SubKwnPhii(I).Text = udtitem.SubKwnPhii(I)
          frminput5.txtDxi(I).Text = udtitem.txtDxi(I)
          frminput5.DenPlastic(I).Text = udtitem.DenPlastic(I)
      Next I
      For I = 1 To 15
```

```
        frminput5.Coordi(I).Text = udtitem.Coordi(I)
        frminput5.InitialValuei(I).Text = udtitem.InitialValuei(I)
    Next I
    For I = 1 To 14
        frminput5.EleNodeDatai(I).Text = udtitem.EleNodeDatai(I)
        frminput5.EleNodeDataj(I).Text = udtitem.EleNodeDataj(I)
        frminput5.EleMatli(I).Text = udtitem.EleMatli(I)
    Next I
    frminput5.DenFood.Text = udtitem.DenFood:  frminput5.txtSurArea.Text = udtitem.txtSurArea
    frminput5.PartitionC.Text = udtitem.PartitionC:  frminput5.WtPlasticS.Text = udtitem.WtPlasticS
    frminput5.VoFoodS.Text = udtitem.VoFoodS:  frminput5.ConcSubIni.Text = udtitem.ConcSubIni
    frminput5.ConcSubVial.Text = udtitem.ConcSubVial
    frminput5.DeltaTimei.Text = udtitem.DeltaTimei
    Close #1
  frminput5.txtCorelayer.SetFocus
  frminput5.Caption = dlgCommonDialog.Filename '& " - Migration Program"
  OpenErrHandler:
End Sub

Private Sub mnuFileSave_Click()
Const conbtns = vbYesNoCancel + vbExclamation + vbDefaultButton3 + vbApplicationModal
Dim udtitem As itemstruc
Dim Filename As String
  On Error GoTo SaveErrHandler
  dlgCommonDialog.CancelError = True
  dlgCommonDialog.Flags = cdlOFNFileMustExist + cdlOFNOverwritePrompt
  dlgCommonDialog.Filter = "Names data files(*.mig)|*.mig|All Files (*.*)|*.*"
  dlgCommonDialog.ShowSave
  Open dlgCommonDialog.Filename For Random As #1 Len = Len(udtitem)
  udtitem.NumNodesi = Val(frminput5.NumNodesi.Text)
  udtitem.NumElei = Val(frminput5.NumElei.Text)
  udtitem.NumMatlSetsi = Val(frminput5.NumMatlSetsi.Text)
  udtitem.NumKwnPhii = Val(frminput5.NumKwnPhii.Text)
  udtitem.txtCorelayer = Val(frminput5.txtCorelayer.Text)
  udtitem.txtOutlayer = Val(frminput5.txtOutlayer.Text)
  udtitem.txtTotalThick = Val(frminput5.txtTotalThick.Text)
  udtitem.txtExtralayer = Val(frminput5.txtExtralayer.Text)
  udtitem.txtTotalMil = Val(frminput5.txtTotalMil.Text)
  For J = 1 To 3
    udtitem.SubKwnPhii(J) = Val(frminput5.SubKwnPhii(J).Text)
    udtitem.txtDxi(J) = Val(frminput5.txtDxi(J).Text)
    udtitem.DenPlastic(J) = Val(frminput5.DenPlastic(J).Text)
  Next J
  For J = 1 To 15
     udtitem.Coordi(J) = Val(frminput5.Coordi(J).Text)
     udtitem.InitialValuei(J) = Val(frminput5.InitialValuei(J).Text)
  Next J
  For J = 1 To 14
      udtitem.EleNodeDatai(J) = Val(frminput5.EleNodeDatai(J).Text)
      udtitem.EleNodeDataj(J) = Val(frminput5.EleNodeDataj(J).Text)
      udtitem.EleMatli(J) = Val(frminput5.EleMatli(J).Text)
  Next J
  udtitem.DenFood = Val(frminput5.DenFood.Text):
  udtitem.txtSurArea = Val(frminput5.txtSurArea.Text)
  udtitem.PartitionC = Val(frminput5.PartitionC.Text)
  udtitem.WtPlasticS = Val(frminput5.WtPlasticS.Text)
```

```
    udtitem.VoFoodS = Val(frminput5.VoFoodS.Text)
    udtitem.ConcSubIni = Val(frminput5.ConcSubIni.Text)
    udtitem.ConcSubVial = Val(frminput5.ConcSubVial.Text)
    udtitem.DeltaTimei = Val(frminput5.DeltaTimei.Text)
    Put #1, 97, udtitem
    Close #1
    frminput5.txtCorelayer.SetFocus
    frminput5.Caption = dlgCommonDialog.Filename '& " - Shelf Life program"
    SaveErrHandler:
    blnCancelSave = True
    Exit Sub
End Sub

Private Sub mnuFileNew_Click()
    frminput5.Picturein2.Cls
    frminput5.txtCorelayer.Text = "":     frminput5.txtOutlayer.Text = ""
    frminput5.txtTotalThick.Text = "":     frminput5.txtExtralayer.Text = ""
    frminput5.txtTotalMil.Text = "":     frminput5.NumNodesi.Text = ""
    frminput5.NumElei.Text = "":     frminput5.NumMatlSetsi.Text = ""
    frminput5.NumKwnPhii.Text = "":     frminput5.SubKwnPhii(1).Text = ""
    frminput5.SubKwnPhii(2).Text = "0":     frminput5.SubKwnPhii(3).Text = "0"
    frminput5.txtDxi(1).Text = "":     frminput5.txtDxi(2).Text = "0"
    frminput5.txtDxi(3).Text = "0":     frminput5.DenPlastic(1).Text = ""
    frminput5.DenPlastic(2).Text = "0":     frminput5.DenPlastic(3).Text = "0"
    frminput5.Coordi(1).Text = "0"
    For I = 2 To 15: frminput5.Coordi(I).Text = "": Next I
    frminput5.InitialValuei(1).Text = "100"
    For J = 2 To 15: frminput5.InitialValuei(J).Text = "": Next J
    For I = 1 To 14: frminput5.EleNodeDatai(I).Text = I: Next I
    For J = 1 To 14: frminput5.EleNodeDataj(J).Text = J + 1: Next J
    For I = 2 To 14: frminput5.EleMatli(I).Text = "1": Next I
    frminput5.VoPlastic.Text = "":     frminput5.txtSurArea.Text = ""
    frminput5.DenFood.Text = "":     frminput5.PartitionC.Text = "0"
    frminput5.WtPlasticS.Text = "":     frminput5.VoFoodS.Text = ""
    frminput5.ConcSubIni.Text = "":     frminput5.ConcSubVial.Text = ""
    frminput5.DeltaTimei.Text = "900":     frminput5.WtFoodS.Text = ""
    frminput5.OptReCalc1 = False:     frminput5.OptReCalc2 = False
    frminput5.optcmdCalc = False:     frminput5.PartFactor.Text = ""
    frminput5.NumTimeStepso.Text = " " & 0:     frminput5.txtTimeCalc.Text = 0
    frminput5.txtTime.Text = " " & 0:     frminput5.TimeHour.Caption = 0
    frminput5.MigPercent.Text = " ":     frminput5.MigPpm.Text = " "
    For I = 1 To 15: frminput5.Text1(I).Text = " ": Next I
    frminput5.Pictureout.Cls
    frminput5.txtCorelayer.SetFocus
 End Sub

Private Sub mnuOpenResults_Click()
Const conbtns = vbYesNoCancel + vbExclamation + vbDefaultButton3 + vbApplicationModal
Dim udtitem As itemstruc
Dim Filename As String
    On Error GoTo OpenErrHandler
    dlgCommonDialog.CancelError = True
    dlgCommonDialog.Flags = cdlOFNFileMustExist + cdlOFNOverwritePrompt
    dlgCommonDialog.Filter = "Names data files(*.tst)|*.tst|All Files (*.*)|*.*"
    dlgCommonDialog.Filename = ""
    dlgCommonDialog.ShowOpen
```

```
  Open dlgCommonDialog.Filename For Random As #1 Len = Len(udtitem)
  Get #1, 36, udtitem
     For K = 1 To 18
       frminput5.AnyTimeT(K).Text = udtitem.AnyTimeT(K)
       frminput5.AnyConc(K).Text = udtitem.AnyConc(K)
    Next K
  Close #1
  frminput5.AnyTimeT(1).SetFocus
  frminput5.Caption = dlgCommonDialog.Filename '& " - Migration Program"
  OpenErrHandler:
End Sub

Private Sub mnuSaveResults_Click()
Const conbtns = vbYesNoCancel + vbExclamation + vbDefaultButton3 + vbApplicationModal
Dim udtitem As itemstruc
Dim Filename As String
  On Error GoTo SaveErrHandler
  dlgCommonDialog.CancelError = True
  dlgCommonDialog.Flags = cdlOFNFileMustExist + cdlOFNOverwritePrompt
  dlgCommonDialog.Filter = "Names data files(*.tst)|*.tst|All Files (*.*)|*.*"
  dlgCommonDialog.ShowSave
  Open dlgCommonDialog.Filename For Random As #1 Len = Len(udtitem)
  For K = 1 To 18
    udtitem.AnyTimeT(K) = Val(frminput5.AnyTimeT(K).Text)
    udtitem.AnyConc(K) = Val(frminput5.AnyConc(K).Text)
  Next K
  Put #1, 36, udtitem
  Close #1
  frminput5.AnyTimeT(1).SetFocus
  frminput5.Caption = dlgCommonDialog.Filename '& " - Shelf Life program"
  SaveErrHandler:
  blnCancelSave = True
  Exit Sub
End Sub

Private Sub mnuNewResults_Click()
  For K = 1 To 18
    frminput5.AnyTimeT(K).Text = ""
    frminput5.AnyConc(K).Text = ""
  Next K
    frminput5.MaxCVial.Text = ""
    frminput5.NumMigData.Text = ""
    frminput5.TimeCalcSim.Text = ""
End Sub
```

"frmInput5" Code

```
Private Declare Function OSWinHelp% Lib "user32" Alias "WinHelpA" (ByVal hwnd&, ByVal
HelpFile$, ByVal wCommand%, dwData As Any)

Private Sub cmdTotalThick_Click()
Thcl = Val(txtCorelayer.Text):  Thol = Val(txtOutlayer.Text):  Thxl = Val(txtExtralayer.Text)
TTh = Thcl + Thol + Thxl
txtTotalThick = TTh:  chkCoord.Text = TTh
txtTotalMil = Format(TTh / 2.54 * 1000, ".#")
End Sub

Private Sub cmdUnitGrid_Click()
For I = 2 To 15: Coordi(I) = "": Next I
For N = 2 To NumNodesi
Coordi(N) = Format(Val(txtTotalThick.Text) / NumElei * (N - 1), ".######")
Next N
SubKwnPhii(1).Text = NumNodesi
Clb = vbBlack:  Clr = vbRed:  Clbl = vbBlue:  Cly = vbYellow
Picturein2.Cls
'Total thickness = 7560
'X-axis;1500 + 7560 + 2940 = 12000, 'Y-axiz;1000 + 4000 + 1000 = 6000
Picturein2.ScaleWidth = 12000:  Picturein2.ScaleHeight = 6000
Picturein2.DrawWidth = 2 'draw X-Y axis and boundary
Picturein2.Line (1500, 6000 - 1000)-(11500, 6000 - 1000), Clb    'X axis
Picturein2.Line (1500, 6000 - 1000)-(1500, 6000 - 5000), Clb    'Y axis
Picturein2.Line (9060, 6000 - 1000)-(9060, 6000 - 5000), Clb    'boundary between palstic & food
Picturein2.CurrentX = 4500:  Picturein2.CurrentY = 750:  Picturein2.Print "PLASTIC"
Picturein2.CurrentX = 10000:  Picturein2.CurrentY = 750:  Picturein2.Print "FOOD"

Thcl = Val(txtCorelayer.Text):  Thol = Val(txtOutlayer.Text):  Thxl = Val(txtExtralayer.Text)
TTh = Thcl + Thol + Thxl
If Thcl <= 0 Then
Picturein1.Line (1500, 6000 - 1000)-(1500, 6000 - 5000), Clbl    'Y axis
Else
Picturein2.DrawWidth = 1.5
Picturein2.Line (1500 + 7560 * Thcl / TTh, 6000 - 1000)-(1500 + 7560 * Thcl / TTh, 6000 - 4500), Clbl
Thnel = Val(NumElei.Text)
TThcoord = 7560 / TTh
K = 1
Picturein2.DrawWidth = 1
For I = 1 To Thnel + 1
Picturein2.Line (1500 + Val(Coordi(K).Text) * TThcoord, 6000 - 1000)-(1500 + Val(Coordi(K).Text) *
TThcoord, 6000 - 4000), Clr    'Node line in red.
K = K + 1
Next I
Picturein2.CurrentX = 1000 + 7560 * Thcl / 2 / TTh:  Picturein2.CurrentY = 1500
Picturein2.Print "[Core]"
Picturein2.CurrentX = 1000 + 7560 * Thcl / TTh + 7560 * Thol / 2 / TTh:  Picturein2.CurrentY = 1500
Picturein2.Print "[Outer]"
Picturein2.CurrentX = 1450:  Picturein2.CurrentY = 5100:  Picturein2.Print "0"
Picturein2.CurrentX = 1100 + 7560 * Thcl / TTh:  Picturein2.CurrentY = 5100:  Picturein2.Print Thcl
Picturein2.CurrentX = 1100 + 7560:  Picturein2.CurrentY = 5100:  Picturein2.Print Thcl + Thol
Picturein2.CurrentX = 1150:  Picturein2.CurrentY = 4800:  Picturein2.Print "0"
End If
End Sub
```

```
Private Sub txtOutlayer_Change()
Thcl = Val(txtCorelayer.Text):  Thol = Val(txtOutlayer.Text):   Thxl = Val(txtExtralayer.Text)
TTh = Thcl + Thol + Thxl
txtTotalThick = TTh:  chkCoord.Text = TTh
txtTotalMil = Format(TTh / 2.54 * 1000, ".#")
End Sub

Private Sub NumElei_Change()
NumNodesi.Text = Val(NumElei.Text) + 1:    chkNumNodes.Text = Val(NumElei.Text) + 1
End Sub

Private Sub Form_Load()
ComboNum.AddItem "1":  ComboNum.AddItem "2":  ComboNum.AddItem "3"
ComboNum.AddItem "4": ComboNum.AddItem "5":  ComboNum.AddItem "6": ComboNum.AddItem "7"
End Sub

Private Sub cmdDiff_click()
Dim SX As Single, SXS As Single, SY As Single, SXY As Single, SCF As Single
SX = 0:  SXS = 0:  SY = 0:  SXY = 0
SCF = (0.000001 * DVoLiq / DSurArea) / (DIniConc * 0.000001 * DDenPlas)
For N = 1 To ComboNum
SX = SX + DTimeSqrt(N) * 60
SXS = SXS + (DTimeSqrt(N) * 60) ^ 2
SY = SY + ConcRes(N) * SCF
SXY = SXY + DTimeSqrt(N) * 60 * (ConcRes(N) * SCF)
Next N
A1 = (ComboNum * SXY - SX * SY) / (ComboNum * SXS - SX ^ 2)
A0 = SY / ComboNum - A1 * SX / ComboNum
txtDiff.Text = Format(A1 ^ 2 * 3.1416 / 4, "#.##E+")
SS = 0:  SSR = 0
For N = 1 To ComboNum
SS = SS + (ConcRes(N) * SCF - SY / ComboNum) ^ 2
SSR = SSR + (ConcRes(N) * SCF - A0 - A1 * DTimeSqrt(N) * 60) ^ 2
Next N
RSQ = (SS - SSR) / SS
txtRsq.Text = Format(RSQ, ".####")
'to draw a straight line
GSX = 0:  GSXS = 0:  GSY = 0:  GSXY = 0
For N = 1 To ComboNum
GSX = GSX + DTimeSqrt(N)
GSXS = GSXS + DTimeSqrt(N) ^ 2
GSY = GSY + ConcRes(N)
GSXY = GSXY + DTimeSqrt(N) * ConcRes(N)
Next N
GA1 = (ComboNum * GSXY - GSX * GSY) / (ComboNum * GSXS - GSX ^ 2)
Clb = vbBlack:  Clr = vbRed:  Clbl = vbBlue:  Cly = vbYellow:  Clm = vbMagenta:  Clc = vbCyan
Dim LineColor As Long
LineColor = RGB(175, 175, 250)
PictureDiff.Cls
PictureDiff.ScaleWidth = 12000:  PictureDiff.ScaleHeight = 6000
'x-gridlines
PictureDiff.DrawWidth = 1
PictureDiff.Line (0, 200)-(12000, 200), LineColor:  PictureDiff.Line (0, 1000)-(12000, 1000), LineColor
PictureDiff.Line (0, 1800)-(12000, 1800), LineColor:  PictureDiff.Line (0, 2600)-(12000, 2600), LineColor
PictureDiff.Line (0, 3400)-(12000, 3400), LineColor:  PictureDiff.Line (0, 4200)-(12000, 4200), LineColor
PictureDiff.Line (0, 5000)-(12000, 5000), LineColor:  PictureDiff.Line (0, 5800)-(12000, 5800), LineColor
```

```
PictureDiff.Line (0, 200)-(12000, 200), LineColor:  PictureDiff.Line (0, 600)-(12000, 600), LineColor
PictureDiff.Line (0, 1400)-(12000, 1400), LineColor:  PictureDiff.Line (0, 2200)-(12000, 2200), LineColor
PictureDiff.Line (0, 3000)-(12000, 3000), LineColor:  PictureDiff.Line (0, 3800)-(12000, 3800), LineColor
PictureDiff.Line (0, 4600)-(12000, 4600), LineColor:  PictureDiff.Line (0, 5400)-(12000, 5400), LineColor
'y-gridlines
PictureDiff.Line (50, 0)-(50, 6000), LineColor:  PictureDiff.Line (1750, 0)-(1750, 6000), LineColor
PictureDiff.Line (3450, 0)-(3450, 6000), LineColor:  PictureDiff.Line (5150, 0)-(5150, 6000), LineColor
PictureDiff.Line (6850, 0)-(6850, 6000), LineColor:  PictureDiff.Line (8550, 0)-(8550, 6000), LineColor
PictureDiff.Line (10250, 0)-(10250, 6000), LineColor:  PictureDiff.Line (11950, 0)-(11950, 6000),
LineColor:  PictureDiff.Line (50, 0)-(50, 6000), LineColor:  PictureDiff.Line (900, 0)-(900, 6000),
LineColor:  PictureDiff.Line (2600, 0)-(2600, 6000), LineColor:  PictureDiff.Line (4300, 0)-(4300, 6000),
LineColor:  PictureDiff.Line (6000, 0)-(6000, 6000), LineColor:  PictureDiff.Line (7700, 0)-(7700, 6000),
LineColor:  PictureDiff.Line (9400, 0)-(9400, 6000), LineColor:  PictureDiff.Line (11100, 0)-(11100,
6000), LineColor
PictureDiff.DrawWidth = 1 'draw X-Y axis and boundary
PictureDiff.Line (1750, 6000 - 1000)-(11100, 6000 - 1000), Clb    'X axis
PictureDiff.Line (1750, 6000 - 1000)-(1750, 6000 - 5400), Clb    'Y axis
PictureDiff.CurrentX = 1700:  PictureDiff.CurrentY = 5100:  PictureDiff.Print "0"
PictureDiff.CurrentX = 1400:  PictureDiff.CurrentY = 4800:  PictureDiff.Print "0"
PictureDiff.DrawWidth = 1
PictureDiff.Line (1750, 1000)-(1850, 1000):  PictureDiff.Line (1750, 1800)-(1850, 1800)
PictureDiff.Line (1750, 2600)-(1850, 2600):  PictureDiff.Line (1750, 3400)-(1850, 3400)
PictureDiff.Line (1750, 4200)-(1850, 4200)
PictureDiff.CurrentX = 800:  PictureDiff.CurrentY = 900:  PictureDiff.Print "100%"
PictureDiff.CurrentX = 950:  PictureDiff.CurrentY = 1700:  PictureDiff.Print "80%"
PictureDiff.CurrentX = 950:  PictureDiff.CurrentY = 2500:  PictureDiff.Print "60%"
PictureDiff.CurrentX = 950:  PictureDiff.CurrentY = 3300:  PictureDiff.Print "40%"
PictureDiff.CurrentX = 950:  PictureDiff.CurrentY = 4100:  PictureDiff.Print "20%"
PictureDiff.CurrentX = 400:  PictureDiff.CurrentY = 500:  PictureDiff.Print "Mt/Me"
If DTimeSqrt(ComboNum) <= 3 Then       DMaxXax = 3
ElseIf DTimeSqrt(ComboNum) <= 4 Then      DMaxXax = 4
ElseIf DTimeSqrt(ComboNum) <= 5 Then     DMaxXax = 5
ElseIf DTimeSqrt(ComboNum) <= 6 Then     DMaxXax = 6
ElseIf DTimeSqrt(ComboNum) <= 7 Then     DMaxXax = 7
ElseIf DTimeSqrt(ComboNum) <= 8 Then     DMaxXax = 8
ElseIf DTimeSqrt(ComboNum) <= 9 Then     DMaxXax = 9
ElseIf DTimeSqrt(ComboNum) <= 10 Then      DMaxXax = 10
ElseIf DTimeSqrt(ComboNum) <= 12 Then     DMaxXax = 12
ElseIf DTimeSqrt(ComboNum) <= 15 Then     DMaxXax = 15
ElseIf DTimeSqrt(ComboNum) <= 18 Then     DMaxXax = 18
ElseIf DTimeSqrt(ComboNum) <= 20 Then     DMaxXax = 20
ElseIf DTimeSqrt(ComboNum) <= 25 Then     DMaxXax = 25
ElseIf DTimeSqrt(ComboNum) <= 30 Then      DMaxXax = 30
ElseIf DTimeSqrt(ComboNum) <= 40 Then      DMaxXax = 40
ElseIf DTimeSqrt(ComboNum) <= 50 Then      DMaxXax = 50
Else       DMaxXax = 100
End If
PictureDiff.DrawWidth = 1
PictureDiff.Line (10250, 4930)-(10250, 5000):  PictureDiff.Line (8550, 4930)-(8550, 5000)
PictureDiff.Line (6850, 4930)-(6850, 5000):  PictureDiff.Line (5150, 4930)-(5150, 5000)
PictureDiff.Line (3450, 4930)-(3450, 5000)
PictureDiff.CurrentX = 10000:  PictureDiff.CurrentY = 5100
PictureDiff.Print Format(DMaxXax, "fixed")
PictureDiff.CurrentX = 8300:  PictureDiff.CurrentY = 5100
PictureDiff.Print Format(DMaxXax * 0.8, "fixed")
PictureDiff.CurrentX = 6600:  PictureDiff.CurrentY = 5100
```

```
PictureDiff.Print Format(DMaxXax * 0.6, "fixed")
PictureDiff.CurrentX = 4900:  PictureDiff.CurrentY = 5100
PictureDiff.Print Format(DMaxXax * 0.4, "fixed")
PictureDiff.CurrentX = 3200:  PictureDiff.CurrentY = 5100
PictureDiff.Print Format(DMaxXax * 0.2, "fixed")
PictureDiff.CurrentX = 10650:  PictureDiff.CurrentY = 5100:  PictureDiff.Print " hrs½"
'Test Conc. at each time interval
DMaxMigConc = DIniConc * (DSurArea * DThick * DDenPlas) / DVoLiq
PictureDiff.DrawWidth = 3
For J = 1 To ComboNum
PictureDiff.Line (1750 + DTimeSqrt(J) / DMaxXax * 8500, 5000 - ConcRes(J) / DMaxMigConc * 4000)-
(1780 + DTimeSqrt(J) / DMaxXax * 8500, 5000 - ConcRes(J) / DMaxMigConc * 4000), Clbl
Next J
PictureDiff.DrawWidth = 1
PictureDiff.Line (1750, 5000)-(1750 + (DMaxMigConc / GA1) / DMaxXax * 8500, 1000), Clr
End Sub

Private Sub cmdDrawing_Click()
Clb = vbBlack:  Clr = vbRed:  Clbl = vbBlue:  Cly = vbYellow:  Clm = vbMagenta:  Clc = vbCyan
Dim LineColor As Long
LineColor = RGB(175, 175, 250)
PictureDiff.Cls
PictureDiff.ScaleWidth = 12000: PictureDiff.ScaleHeight = 6000
'x-gridlines
PictureDiff.DrawWidth = 1
PictureDiff.Line (0, 200)-(12000, 200), LineColor: PictureDiff.Line (0, 1000)-(12000, 1000), LineColor
PictureDiff.Line (0, 1800)-(12000, 1800), LineColor: PictureDiff.Line (0, 2600)-(12000, 2600), LineColor
PictureDiff.Line (0, 3400)-(12000, 3400), LineColor: PictureDiff.Line (0, 4200)-(12000, 4200), LineColor
PictureDiff.Line (0, 5000)-(12000, 5000), LineColor: PictureDiff.Line (0, 5800)-(12000, 5800), LineColor
PictureDiff.Line (0, 200)-(12000, 200), LineColor: PictureDiff.Line (0, 600)-(12000, 600), LineColor
PictureDiff.Line (0, 1400)-(12000, 1400), LineColor: PictureDiff.Line (0, 2200)-(12000, 2200), LineColor
PictureDiff.Line (0, 3000)-(12000, 3000), LineColor: PictureDiff.Line (0, 3800)-(12000, 3800), LineColor
PictureDiff.Line (0, 4600)-(12000, 4600), LineColor: PictureDiff.Line (0, 5400)-(12000, 5400), LineColor
'y-gridlines
PictureDiff.Line (50, 0)-(50, 6000), LineColor: PictureDiff.Line (1750, 0)-(1750, 6000), LineColor
PictureDiff.Line (3450, 0)-(3450, 6000), LineColor: PictureDiff.Line (5150, 0)-(5150, 6000), LineColor
PictureDiff.Line (6850, 0)-(6850, 6000), LineColor: PictureDiff.Line (8550, 0)-(8550, 6000), LineColor
PictureDiff.Line (10250, 0)-(10250, 6000), LineColor: PictureDiff.Line (11950, 0)-(11950, 6000),
LineColor
PictureDiff.Line (50, 0)-(50, 6000), LineColor: PictureDiff.Line (900, 0)-(900, 6000), LineColor
PictureDiff.Line (2600, 0)-(2600, 6000), LineColor: PictureDiff.Line (4300, 0)-(4300, 6000), LineColor
PictureDiff.Line (6000, 0)-(6000, 6000), LineColor: PictureDiff.Line (7700, 0)-(7700, 6000), LineColor
PictureDiff.Line (9400, 0)-(9400, 6000), LineColor: PictureDiff.Line (11100, 0)-(11100, 6000), LineColor
PictureDiff.DrawWidth = 1 'draw X-Y axis and boundary
PictureDiff.Line (1750, 6000 - 1000)-(11100, 6000 - 1000), Clb    'X axis
PictureDiff.Line (1750, 6000 - 1000)-(1750, 6000 - 5400), Clb    'Y axis
PictureDiff.CurrentX = 1700: PictureDiff.CurrentY = 5100: PictureDiff.Print "0"
PictureDiff.CurrentX = 1400: PictureDiff.CurrentY = 4800: PictureDiff.Print "0"
PictureDiff.DrawWidth = 1
PictureDiff.Line (1750, 1000)-(1850, 1000): PictureDiff.Line (1750, 1800)-(1850, 1800)
PictureDiff.Line (1750, 2600)-(1850, 2600): PictureDiff.Line (1750, 3400)-(1850, 3400)
PictureDiff.Line (1750, 4200)-(1850, 4200)
PictureDiff.CurrentX = 800: PictureDiff.CurrentY = 900: PictureDiff.Print "100%"
PictureDiff.CurrentX = 950: PictureDiff.CurrentY = 1700: PictureDiff.Print "80%"
PictureDiff.CurrentX = 950: PictureDiff.CurrentY = 2500: PictureDiff.Print "60%"
PictureDiff.CurrentX = 950: PictureDiff.CurrentY = 3300: PictureDiff.Print "40%"
```

```
PictureDiff.CurrentX = 950: PictureDiff.CurrentY = 4100: PictureDiff.Print "20%"
PictureDiff.CurrentX = 400: PictureDiff.CurrentY = 500: PictureDiff.Print "Mt/Me"
If DTimeSqrt(ComboNum) <= 3 Then        DMaxXax = 3
ElseIf DTimeSqrt(ComboNum) <= 4 Then     DMaxXax = 4
ElseIf DTimeSqrt(ComboNum) <= 5 Then     DMaxXax = 5
ElseIf DTimeSqrt(ComboNum) <= 6 Then     DMaxXax = 6
ElseIf DTimeSqrt(ComboNum) <= 7 Then     DMaxXax = 7
ElseIf DTimeSqrt(ComboNum) <= 8 Then     DMaxXax = 8
ElseIf DTimeSqrt(ComboNum) <= 9 Then     DMaxXax = 9
ElseIf DTimeSqrt(ComboNum) <= 10 Then     DMaxXax = 10
ElseIf DTimeSqrt(ComboNum) <= 12 Then     DMaxXax = 12
ElseIf DTimeSqrt(ComboNum) <= 15 Then     DMaxXax = 15
ElseIf DTimeSqrt(ComboNum) <= 18 Then     DMaxXax = 18
ElseIf DTimeSqrt(ComboNum) <= 20 Then     DMaxXax = 20
ElseIf DTimeSqrt(ComboNum) <= 25 Then     DMaxXax = 25
ElseIf DTimeSqrt(ComboNum) <= 30 Then      DMaxXax = 30
ElseIf DTimeSqrt(ComboNum) <= 40 Then      DMaxXax = 40
ElseIf DTimeSqrt(ComboNum) <= 50 Then      DMaxXax = 50
Else      DMaxXax = 100
End If
PictureDiff.DrawWidth = 1
PictureDiff.Line (10250, 4930)-(10250, 5000): PictureDiff.Line (8550, 4930)-(8550, 5000)
PictureDiff.Line (6850, 4930)-(6850, 5000): PictureDiff.Line (5150, 4930)-(5150, 5000)
PictureDiff.Line (3450, 4930)-(3450, 5000)
PictureDiff.CurrentX = 10000: PictureDiff.CurrentY = 5100: PictureDiff.Print Format(DMaxXax, "fixed")
PictureDiff.CurrentX = 8300: PictureDiff.CurrentY = 5100
PictureDiff.Print Format(DMaxXax * 0.8, "fixed")
PictureDiff.CurrentX = 6600: PictureDiff.CurrentY = 5100
PictureDiff.Print Format(DMaxXax * 0.6, "fixed")
PictureDiff.CurrentX = 4900: PictureDiff.CurrentY = 5100
PictureDiff.Print Format(DMaxXax * 0.4, "fixed")
PictureDiff.CurrentX = 3200: PictureDiff.CurrentY = 5100
PictureDiff.Print Format(DMaxXax * 0.2, "fixed")
PictureDiff.CurrentX = 10650: PictureDiff.CurrentY = 5100: PictureDiff.Print " hrs½"
'Test Conc. at each time interval
DMaxMigConc = DIniConc * (DSurArea * DThick * DDenPlas) / DVoLiq
PictureDiff.DrawWidth = 3
For J = 1 To ComboNum
 PictureDiff.Line (1750 + DTimeSqrt(J) / DMaxXax * 8500, 5000 - ConcRes(J) / DMaxMigConc * 4000)-
(1780 + DTimeSqrt(J) / DMaxXax * 8500, 5000 - ConcRes(J) / DMaxMigConc * 4000), Clbl
Next J
End Sub

Private Sub cmdWtPlastic_Click()
WtP = (DenPlastic(1) * Thcl + DenPlastic(2) * Thol + DenPlastic(3) * Thxl) * txtSurArea
WtPlasticS.Text = Format(WtP, ".####")
VoPlastic = Format(txtSurArea * TTh, ".####")
End Sub

Private Sub comWtFoodS_Click()
 WtFoodS = Format(VoFoodS * DenFood, ".####")
End Sub

Private Sub cmdConcSubVial_Click()
ConcSubVial = Format(ConcSubIni * DenPlastic(1) * txtSurArea * txtCorelayer / VoFoodS, ".##")
End Sub
```

```
Private Sub DeltaTimei_Change()
optcmdCalc = False:    OptReCalc1 = False:    OptReCalc2 = False
End Sub

Private Sub optcmdCalc_click()
DeltaTimeHr = Format(DeltaTimei * (1 / 3600), ".##")
Close #1
Open "c:\diffusion.dat" For Output As #1
Write #1, "Lumped"
Write #1, Val(NumNodesi.Text), Val(NumElei.Text), Val(NumMatlSetsi.Text)
Write #1, 0, 0, Val(NumKwnPhii.Text) 'Val(NumKwnSourcei.Text), Val(NumDerivBCi.Text),
For I = 1 To Val(NumMatlSetsi.Text):  Write #1, Val(txtDxi(I).Text), 0, 0, 1:  Next I
For I = 1 To Val(NumNodesi.Text):  Write #1, Val(Coordi(I).Text), Val(InitialValuei(I).Text): Next I
For I = 1 To Val(NumElei.Text)
Write #1, Val(EleNodeDatai(I).Text), Val(EleNodeDataj(I).Text), Val(EleMatli(I).Text)
Next I
For I = 1 To Val(NumKwnPhii.Text):  Write #1, Val(SubKwnPhii(I).Text):  Next I
Write #1, 0.5, Val(DeltaTimei.Text)
Close #1
'Dimension the input data arrays
ReDim MatlValues(Val(NumMatlSetsi.Text), 4) As Single 'Material Values
ReDim Coord(Val(NumNodesi.Text)) As Single   'X Coordinate data
ReDim InitialValue(Val(NumNodesi.Text)) As Single    'Initial values of Phi
ReDim EleNodeData(Val(NumElei.Text), 2) As Integer   'Element node data
ReDim EleMatl(Val(NumElei.Text)) As Integer          'Element material index
ReDim SubKwnSource(1) As Integer
ReDim ValKwnSource(1) As Single
ReDim SubKwnDBC(1) As Integer 'Subscript of the derivative bdy condition
ReDim MValKwnDBC(1) As Single 'M value for the derivative bdy condition
ReDim SValKwnDBC(1) As Single 'S value for the derivative bdy condition
ReDim SubKwnPhi(Val(NumKwnPhii.Text) + 1) As Integer 'Subscripts of the known Phi values.
Open "c: \diffusion.dat" For Input As #1
Input #1, cboType
Input #1, NumNodes, NumEle, NumMatlSets
Input #1, NumKwnSource, NumDerivBC, NumKwnPhi
'Input the basic data
Call InputMatlValues(NumMatlSets, MatlValues())
Call InputCoordData(NumNodes, Coord(), InitialValue())
Call InputNodeData(NumEle, EleNodeData(), EleMatl())
Call InputSourceVal(NumKwnSource, SubKwnSource(), ValKwnSource())
Call InputDerivBdyCd(NumDerivBC, SubKwnDBC(), MValKwnDBC(), SValKwnDBC())
Call InputKwnPhiData(NumKwnPhi, SubKwnPhi())
Input #1, Theta, DeltaTime
Close #1  'End of data input
'Dimension the element matrices
ReDim EFV(NumEleSub)            'Element force vector
ReDim ESM(NumEleSub, NumEleSub) As Single   'Element stiffness matrix
ReDim ECM(NumEleSub, NumEleSub) As Single   'Element capacitance matrix
ReDim EleSub(NumEleSub) As Integer 'Element displacement subscript values
'Dimension the arrays for the global system of equations
Dim NodalValuePerNode As Integer, NumNodalVal As Integer
NodalValuePerNode = 1
NumNodalVal = NumNodes * NodalValuePerNode
Call CalBandWidth1D(NumEle, EleNodeData(), NodalValuePerNode, BandWidth)
ReDim GFV(NumNodalVal)          'Global force vector
ReDim GCM(NumNodalVal, BandWidth)  'Global capacitance matrix
```

```
ReDim GSM(NumNodalVal, BandWidth)  'Global stiffness matrix
ReDim GlobalPhi(NumNodalVal)       'New Phi values
ReDim ProductVector(NumNodalVal)   'Vector used in the solution process
'Build the system of equations element by element
Call ZeroGlobalMatrices(NumNodalVal, BandWidth, GFV(), GCM())
Call ZeroGlobalMatrices(NumNodalVal, BandWidth, GFV(), GCM())
For KK = 1 To NumEle
Call EleStiffMtx1DTime(cboType$, MatlValues!(), EleMatl%(), Coord!(), EleNodeData%(), EFV(),
ESM!(), ECM!(), KK)
Call EleSubscriptValues(KK%, EleNodeData%(), EleSub%())
Call BuildBandedTime(NumEleSub%, EleSub%(), EFV(), ECM!(), ESM!(), GFV(), GCM(), GSM())
Next KK
Call SourceSinkValue(NumKwnSource%, SubKwnSource%(), ValKwnSource!(), GFV())
Call DerivBdyCondition(NumDerivBC%, SubKwnDBC%(), MValKwnDBC!(), SValKwnDBC!(), GSM(),
GFV())
'Modify the [C] and [K] matrices to incorporate the known Phi values
'Convert [C] and [K] to [A] and [P]; Decompose [A]
Call DefineGlobalPhi(NumNodalVal%, InitialValue!(), GlobalPhi())
Call ModifyCandK(NumNodalVal%, BandWidth%, NumKwnPhi%, SubKwnPhi%(), GlobalPhi(), GFV(),
GSM(), GCM())
Call ConvertCandK(NumNodes%, BandWidth%, Theta, DeltaTime, GCM(), GSM(), GFV())
Call ConvertFvector(NumNodes%, Theta, DeltaTime, GFV())
Call DecompBandMatrix(NumNodalVal%, BandWidth%, GCM())
'Calcualtion of Percent Partition from patition coefficient:
If PartitionC = 0 Then
MultiFactor = 1
Else
MultiFactor = 1 / (1 + PartitionC * WtPlasticS / DenFood / VoFoodS)
End If
MaxMigConc = ConcSubVial:  PartFactor.Text = MultiFactor
NumTimeStepso.Text = " " & 0:  txtTimeCalc.Text = 0:  txtTime.Text = " " & 0:  TimeHour.Caption = 0
MigPercent.Text = " ":  MigPpm.Text = " ":  Pictureout.Cls:
For N = 1 To 15:  Text1(N).Text = "":  Next N
optcmdCalc = False:  OptReCalc1 = False:  OptReCalc2 = False
End Sub

Private Sub OptReCalc1_Click()
NumTimeStepso.Text = " " & 0:  txtTimeCalc.Text = 0:  txtTime.Text = " " & 0:  TimeHour.Caption = 0
Call DefineGlobalPhi(NumNodalVal%, InitialValue!(), GlobalPhi())
Call OutputScreen(1, NumNodalVal%, Timet, GlobalPhi())
KK = 1
MigPercent.Text = " ":  MigPpm.Text = " ":  Pictureout.Cls
Open "c: \Diffusionper.dat" For Output As #2
Write #2, DeltaTime * Val(NumTimeStepso.Text) / 3600, Val(MigPercent.Text)
Close #2
Open "c: \Diffusionout.dat" For Output As #3
For I = 1 To NumNodalVal:  Write #3, GlobalPhi(I):  Next I
Close #3
NodalValuePerNode = 1:  NumNodalVal = NumNodes * NodalValuePerNode
'Display the initial values and open the output file
'Area-initial under the line
IniArea = 0
For I = 1 To NumNodes - 1
IniArea = IniArea + (InitialValue(I) + InitialValue(I + 1)) / 2 * (Coord(I + 1) - Coord(I))
Next I
End Sub
```

```
Private Sub OptReCalc2_Click()
NumTimeStepso.Text = " " & 0:  txtTimeCalc.Text = 0
txtTime.Text = " " & 0:  TimeHour.Caption = 0:  MigPercent.Text = " ":  MigPpm.Text = " "
Pictureout.Cls
Call DefineGlobalPhi(NumNodalVal%, InitialValue!(), GlobalPhi())
Call OutputScreen(1, NumNodalVal%, Timet, GlobalPhi())
KK = 1
Open "c: \Diffusionper.dat" For Output As #2
Write #2, DeltaTime * Val(NumTimeStepso.Text) / 3600, Val(MigPercent.Text)
Close #2
Open "c: \Diffusionout.dat" For Output As #3
For I = 1 To NumNodalVal:    Write #3, GlobalPhi(I):   Next I
Close #3
NodalValuePerNode = 1:  NumNodalVal = NumNodes * NodalValuePerNode
IniArea = 0
For I = 1 To NumNodes - 1
IniArea = IniArea + (InitialValue(I) + InitialValue(I + 1)) / 2 * (Coord(I + 1) - Coord(I))
Next I
End Sub

Private Sub DerivBdyCondition(NumDerivBC%, SubKwnDBC%(), MValKwnDBC!(), SValKwnDBC!(),
GSM(), GFV())
'incorporates the derivative boundary condition into the global stiffness matrix and the global force vector
If (NumDerivBC > 0) Then
For I = 1 To NumDerivBC
II = SubKwnDBC(I)
GSM(II, 1) = GSM(II, 1) + MValKwnDBC(I)
GFV(II) = GFV(II) + SValKwnDBC(I)
Next I
End If
End Sub

Private Sub EleStiffMtx1DTime(cboType$, MatlValue!(), EleMatl%(), Coord!(), EleNodeData%(), EFV(),
ESM!(), ECM!(), KK)
'calculates the element stiffness matrix and element force vector for the one-dimensional time problem.
Dim Dx As Single, G As Single, Q As Single, Dt As Single
MM = EleMatl(KK)
Dx = MatlValue(MM, 1):  G = MatlValue(MM, 2):  Q = MatlValue(MM, 3):  Dt = MatlValue(MM, 4)
'Evaluation of the element stiffness matrix
Dim EleLength As Single, DxOverL As Single, GLOverSix As Single
II = EleNodeData%(KK, 1):  JJ = EleNodeData%(KK, 2)
EleLength = Coord(JJ) - Coord(II)
DxOverL = Dx / EleLength:  GLOverSix = G * EleLength / 6
ESM(1, 1) = DxOverL + GLOverSix * 2:  ESM(1, 2) = -DxOverL + GLOverSix
ESM(2, 1) = ESM(1, 2):  ESM(2, 2) = ESM(1, 1)
'Evaluation of the element capacitance matrix ECM!( , )
If cboType$ = "Lumped" Then
ECM(1, 1) = Dt * EleLength / 2:    ECM(1, 2) = 0:    ECM(2, 1) = 0:    ECM(2, 2) = ECM(1, 1)
Else
ECM(1, 1) = Dt * EleLength / 3:    ECM(1, 2) = Dt * EleLength / 6:    ECM(2, 1) = ECM(1, 2)
ECM(2, 2) = ECM(1, 1)
End If
'Evaluation of the element force vector, EFV( )
EFV(1) = Q * EleLength / 2:  EFV(2) = Q * EleLength / 2
End Sub
```

```
Private Sub EleSubscriptValues(KK%, EleNodeData%(), EleSub%())
'calculates the subscripts associated with the element.
EleSub(1) = EleNodeData(KK, 1):  EleSub(2) = EleNodeData(KK, 2)
End Sub

Private Sub InputMatlValues(NumMatlSets%, MatlValues!())
'inputs the material sets
For I = 1 To NumMatlSets
Input #1, MatlValues(I, 1), MatlValues(I, 2), MatlValues(I, 3), MatlValues(I, 4)
Next I
End Sub

Private Sub InputCoordData(NumNodes%, Coord!(), InitialValue!())
'inputs the X coordinate of each node and the initial value of Phi for the x coordinate.
For I = 1 To NumNodes
Input #1, Coord!(I), InitialValue!(I)
Next I
End Sub

Private Sub InputNodeData(NumEle As Integer, EleNodeData() As Integer, EleMatl%())
'inputs the element node numbers and the material control value.
For I = 1 To NumEle
Input #1, EleNodeData(I, 1), EleNodeData(I, 2), EleMatl(I)
Next I
End Sub

Private Sub InputSourceVal(NumKwnSource%, SubKwnSource%(), ValKwnSource!())
'inputs the location and values of the known sources and sinks
For I = 1 To NumKwnSource
Input #1, SubKwnSource(I), ValKwnSource(I)
Next I
End Sub

Private Sub InputDerivBdyCd(NumDerivBC%, SubKwnDBC%(), MValKwnDBC!(), SValKwnDBC!())
'inputs the derivative boundary conditions
For I = 1 To NumDerivBC
Input #1, SubKwnDBC(I), MValKwnDBC(I), SValKwnDBC(I)
Next I
End Sub

Private Sub InputKwnPhiData(NumKwnPhi As Integer, SubKwnPhi() As Integer)
'inputs the subscript value and the numerical value of each known value of phi
For I = 1 To NumKwnPhi
Input #1, SubKwnPhi(I)
Next I
End Sub

Sub SourceSinkValue(NumKwnSource%, SubKwnSource%(), ValKwnSource!(), GFV())
'incorporates the known source or sink values into the global force vector
For I = 1 To NumKwnSource
J = SubKwnSource(I)
GFV(J) = GFV(J) + ValKwnSource(I)
Next I
End Sub
```

```
Private Sub cmdChkSystem_Click()
Clb = vbBlack: Clr = vbRed: Clbl = vbBlue: Cly = vbYellow:
Picturein2.Cls
Picturein2.ScaleWidth = 12000  'start point (1500, 1000)
Picturein2.ScaleHeight = 6000
Picturein2.DrawWidth = 2 'draw X-Y axis and boundary
Picturein2.Line (1500, 6000 - 1000)-(11500, 6000 - 1000), Clb    'X axis
Picturein2.Line (1500, 6000 - 1000)-(1500, 6000 - 5000), Clb    'Y axis
Picturein2.Line (9060, 6000 - 1000)-(9060, 6000 - 5000), Clb   'boundary between palstic & food
Picturein2.CurrentX = 4500: Picturein2.CurrentY = 750: Picturein2.Print "PLASTIC"
Picturein2.CurrentX = 10000: Picturein2.CurrentY = 750: Picturein2.Print "FOOD"
Thcl = Val(txtCorelayer.Text): Thol = Val(txtOutlayer.Text): Thxl = Val(txtExtralayer.Text)
TTh = Thcl + Thol + Thxl
If Thcl <= 0 Then
Picturein1.Line (1500, 6000 - 1000)-(1500, 6000 - 5000), Clbl    'Y axis
Else
Picturein2.DrawWidth = 1.5
Picturein2.Line (1500 + 7560 * Thcl / TTh, 6000 - 1000)-(1500 + 7560 * Thcl / TTh, 6000 - 4500), Clbl
'boundary between core layer & outer layer
Thnel = Val(NumElei.Text): TThcoord = 7560 / TTh
K = 1
Picturein2.DrawWidth = 1
For I = 1 To Thnel + 1
Picturein2.Line (1500 + Val(Coordi(K).Text) * TThcoord, 6000 - 1000)-(1500 + Val(Coordi(K).Text) *
TThcoord, 6000 - 4000), Clr    'Node line in red.
K = K + 1
Next I
Picturein2.CurrentX = 1000 + 7560 * Thcl / 2 / TTh: Picturein2.CurrentY = 1500
Picturein2.Print "[Core]"
Picturein2.CurrentX = 1000 + 7560 * Thcl / TTh + 7560 * Thol / 2 / TTh: Picturein2.CurrentY = 1500
Picturein2.Print "[Outer]"
Picturein2.CurrentX = 1450: Picturein2.CurrentY = 5100: Picturein2.Print "0"
Picturein2.CurrentX = 1100 + 7560 * Thcl / TTh: Picturein2.CurrentY = 5100: Picturein2.Print Thcl
Picturein2.CurrentX = 1100 + 7560: Picturein2.CurrentY = 5100: Picturein2.Print Thcl + Thol
Picturein2.CurrentX = 1150: Picturein2.CurrentY = 4800: Picturein2.Print "0"
Picturein2.CurrentX = 700: Picturein2.CurrentY = 6000 - (4000 * 0.7 + 1100)
Picturein2.Print InitialValuei(1)
Picturein2.DrawWidth = 4  ' Initial Conc. at each node
Thnod = Thnel + 1
For J = 1 To Thnod
Picturein2.Line (1450 + Val(Coordi(J).Text) / TTh * 7560, 6000 - (Val(InitialValuei(J).Text) /
Val(InitialValuei(1).Text) * 4000 * 0.7 + 1000))-(1550 + Val(Coordi(J).Text) / TTh * 7560, 6000 -
(Val(InitialValuei(J).Text) / Val(InitialValuei(1).Text) * 4000 * 0.7 + 1000)), Cly
Next J
Picturein2.DrawWidth = 2 ' Connecting Conc. each other by Y-line
For I = 1 To Thnel
Picturein2.Line (1500 + Val(Coordi(I).Text) / TTh * 7560, 6000 - (Val(InitialValuei(I).Text) /
Val(InitialValuei(1).Text) * 4000 * 0.7 + 1000))-(1500 + Val(Coordi(I + 1).Text) / TTh * 7560, 6000 -
(Val(InitialValuei(I + 1).Text) / Val(InitialValuei(1).Text) * 4000 * 0.7 + 1000)), Cly
Next I
End If
End Sub

Private Sub cmdNext_click()
Clb = vbBlack: Clr = vbRed: Clbl = vbBlue: Cly = vbYellow: Clm = vbMagenta
Pictureout.Cls
```

```
Pictureout.ScaleWidth = 12000:  Pictureout.ScaleHeight = 6000
Pictureout.DrawWidth = 2 'draw X-Y axis and boundary
Pictureout.Line (1500, 6000 - 1000)-(11500, 6000 - 1000), Clb    'X axis
Pictureout.Line (1500, 6000 - 1000)-(1500, 6000 - 5000), Clb    'Y axis
Pictureout.Line (9060, 6000 - 1000)-(9060, 6000 - 5000), Clb    'boundary between palstic & food
Pictureout.CurrentX = 4500:  Pictureout.CurrentY = 750:  Pictureout.Print "PLASTIC"
Pictureout.CurrentX = 10000:  Pictureout.CurrentY = 750:  Pictureout.Print "FOOD"
Pictureout.DrawWidth = 1.5
Pictureout.Line (1500 + 7560 * Thcl / TTh, 6000 - 1000)-(1500 + 7560 * Thcl / TTh, 6000 - 4500), Clbl
'boundary between core layer & outer layer
K = 1
Pictureout.DrawWidth = 1
For I = 1 To Thnel + 1
Pictureout.Line (1500 + Coord(K) * TThcoord, 6000 - 1000)-(1500 + Coord(K) * TThcoord, 6000 - 4000),
Clr    'Node line in red.
K = K + 1
Next I
Pictureout.CurrentX = 1000 + 7560 * Thcl / 2 / TTh:  Pictureout.CurrentY = 1500
Pictureout.Print "[Core]"
Pictureout.CurrentX = 1000 + 7560 * Thcl / TTh + 7560 * Thol / 2 / TTh:  Pictureout.CurrentY = 1500
Pictureout.Print "[Outer]"
Pictureout.CurrentX = 1450:  Pictureout.CurrentY = 5100:  Pictureout.Print "0"
Pictureout.CurrentX = 1100 + 7560 * Thcl / TTh:  Pictureout.CurrentY = 5100:  Pictureout.Print Thcl
Pictureout.CurrentX = 1100 + 7560:  Pictureout.CurrentY = 5100:  Pictureout.Print Thcl + Thol
Pictureout.CurrentX = 1150:  Pictureout.CurrentY = 4800:  Pictureout.Print "0"
Pictureout.CurrentX = 800:  Pictureout.CurrentY = 6000 - (4000 * 0.7 + 1100)
Pictureout.Print InitialValue(1)
Pictureout.DrawWidth = 3
For J = 1 To NumNodes 'Thnod
Pictureout.Line (1450 + Coord(J) / TTh * 7560, 6000 - (InitialValue(J) / InitialValue(1) * 4000 * 0.7 +
1000))-(1550 + Coord(J) / TTh * 7560, 6000 - (InitialValue(J) / InitialValue(1) * 4000 * 0.7 + 1000)), Cly
Next J
'Connecting Conc. each other by Y-line
Pictureout.DrawWidth = 1
For I = 1 To NumEle 'Thnel
Pictureout.Line (1500 + Coord(I) / TTh * 7560, 6000 - (InitialValue(I) / InitialValue(1) * 4000 * 0.7 +
1000))-(1500 + Coord(I + 1) / TTh * 7560, 6000 - (InitialValue(I + 1) / InitialValue(1) * 4000 * 0.7 +
1000)), Clm
Next I
'Do the solution in time steps.
KK = KK + 1
Timet = 0
'For KK = 1 To NumTimeSteps Step 1
Timet = Timet + DeltaTime
Call MultpyBandMatrix(NumNodalVal%, BandWidth%, GSM(), GlobalPhi(), ProductVector())
For I = 1 To NumNodalVal%
ProductVector(I) = ProductVector(I) + GFV(I)
Next I
Call SolveBandMatrix(NumNodalVal%, BandWidth%, GlobalPhi(), ProductVector(), GCM())
Call OutputScreen(KK, NumNodalVal%, Timet, GlobalPhi())
txtTime.Text = " " & Format(DeltaTime * (KK - 1), "#,###,###,###")
NumTimeStepso.Text = " " & KK - 1 'NumTimeSteps
TimeHour.Caption = Format(DeltaTime * (KK - 1) / 3600, "fixed")
'Connecting Conc. each other by Y-line
Pictureout.DrawWidth = 2
For I = 1 To NumEle
```

```
Pictureout.Line (1500 + Coord(I) / TTh * 7560, 6000 - ((GlobalPhi(I) + (InitialValue(I) - GlobalPhi(I)) * (1
- MultiFactor)) / InitialValue(1) * 4000 * 0.7 + 1000))-(1500 + Coord(I + 1) / TTh * 7560, 6000 -
((GlobalPhi(I + 1) + (InitialValue(I + 1) - GlobalPhi(I + 1)) * (1 - MultiFactor)) / InitialValue(1) * 4000 *
0.7 + 1000)), Cly
Next I
'Calculation of the migrated amount
InoArea = 0
For I = 1 To NumNodes - 1
InoArea = InoArea + ((GlobalPhi(I) + (InitialValue(I) - GlobalPhi(I)) * (1 - MultiFactor)) + (GlobalPhi(I +
1) + (InitialValue(I + 1) - GlobalPhi(I + 1)) * (1 - MultiFactor))) / 2 * (Coord(I + 1) - Coord(I))
Next I
MigPercent = (IniArea - InoArea) / IniArea * 100
MigPpm = Format(MigPercent * ConcSubVial / 100, "fixed")
MigPercent = Format(MigPercent, "##.##")

Open "c: \Diffusionper.dat" For Append As #2
Write #2, DeltaTime * Val(NumTimeStepso.Text) / 3600, Val(MigPercent.Text)
Close #2
Open "c: \Diffusionout.dat" For Append As #3
For I = 1 To NumNodalVal
Write #3, GlobalPhi(I)
Next I
Close #3
End Sub

Private Sub cmdTimeCalc_Click()
Clb = vbBlack:  Clr = vbRed:  Clbl = vbBlue:  Cly = vbYellow:  Clm = vbMagenta
Pictureout.Cls
Pictureout.ScaleWidth = 12000:  Pictureout.ScaleHeight = 6000
Pictureout.DrawWidth = 2 'draw X-Y axis and boundary
Pictureout.Line (1500, 6000 - 1000)-(11500, 6000 - 1000), Clb    'X axis
Pictureout.Line (1500, 6000 - 1000)-(1500, 6000 - 5000), Clb    'Y axis
Pictureout.Line (9060, 6000 - 1000)-(9060, 6000 - 5000), Clb    'boundary between palstic & food
Pictureout.CurrentX = 4500:  Pictureout.CurrentY = 750:  Pictureout.Print "PLASTIC"
Pictureout.CurrentX = 10000:  Pictureout.CurrentY = 750:  Pictureout.Print "FOOD"
Pictureout.DrawWidth = 1.5
Pictureout.Line (1500 + 7560 * Thcl / TTh, 6000 - 1000)-(1500 + 7560 * Thcl / TTh, 6000 - 4500), Clbl
'boundary between core layer & outer layer
K = 1
Pictureout.DrawWidth = 1
For I = 1 To Thnel + 1
Pictureout.Line (1500 + Coord(K) * TThcoord, 6000 - 1000)-(1500 + Coord(K) * TThcoord, 6000 - 4000),
Clr    'Node line in red.
K = K + 1
Next I
Pictureout.CurrentX = 1000 + 7560 * Thcl / 2 / TTh:  Pictureout.CurrentY = 1500
Pictureout.Print "[Core]"
Pictureout.CurrentX = 1000 + 7560 * Thcl / TTh + 7560 * Thol / 2 / TTh:  Pictureout.CurrentY = 1500
Pictureout.Print "[Outer]"
Pictureout.CurrentX = 1450:  Pictureout.CurrentY = 5100:  Pictureout.Print "0"
Pictureout.CurrentX = 1100 + 7560 * Thcl / TTh:  Pictureout.CurrentY = 5100:  Pictureout.Print Thcl
Pictureout.CurrentX = 1100 + 7560:  Pictureout.CurrentY = 5100:  Pictureout.Print Thcl + Thol
Pictureout.CurrentX = 1150:  Pictureout.CurrentY = 4800:  Pictureout.Print "0"
Pictureout.CurrentX = 800:  Pictureout.CurrentY = 6000 - (4000 * 0.7 + 1100)
Pictureout.Print InitialValue(1)
```

```
' Initial Conc. at each node.
Pictureout.DrawWidth = 3
For J = 1 To NumNodes 'Thnod
Pictureout.Line (1450 + Coord(J) / TTh * 7560, 6000 - (InitialValue(J) / InitialValue(1) * 4000 * 0.7 +
1000))-(1550 + Coord(J) / TTh * 7560, 6000 - (InitialValue(J) / InitialValue(1) * 4000 * 0.7 + 1000)), Cly
Next J
'Connecting Conc. each other by Y-line
Pictureout.DrawWidth = 1
For I = 1 To NumEle 'Thnel
Pictureout.Line (1500 + Coord(I) / TTh * 7560, 6000 - (InitialValue(I) / InitialValue(1) * 4000 * 0.7 +
1000))-(1500 + Coord(I + 1) / TTh * 7560, 6000 - (InitialValue(I + 1) / InitialValue(1) * 4000 * 0.7 +
1000)), Clm
Next I
'Do the solution in time durations.
Call DefineGlobalPhi(NumNodalVal%, InitialValue!(), GlobalPhi())
KJ = 1
Timet = 0
NumTimeSteps = Val(txtTimeCalc.Text) * 3600 / DeltaTime
For KJ = 1 To NumTimeSteps Step 1
Timet = Timet + DeltaTime
Call MultpyBandMatrix(NumNodalVal%, BandWidth%, GSM(), GlobalPhi(), ProductVector())
For I = 1 To NumNodalVal%
ProductVector(I) = ProductVector(I) + GFV(I)
Next I
Call SolveBandMatrix(NumNodalVal%, BandWidth%, GlobalPhi(), ProductVector(), GCM())
Call OutputScreen(KJ, NumNodalVal%, Timet, GlobalPhi())
Next KJ
txtTime.Text = " " & Format(DeltaTime * NumTimeSteps, "#,###,###,###")
TimeHour.Caption = Format(DeltaTime * NumTimeSteps / 3600, "fixed")
Pictureout.DrawWidth = 2
For I = 1 To NumEle
Pictureout.Line (1500 + Coord(I) / TTh * 7560, 6000 - ((GlobalPhi(I) + (InitialValue(I) - GlobalPhi(I)) * (1
- MultiFactor)) / InitialValue(1) * 4000 * 0.7 + 1000))-(1500 + Coord(I + 1) / TTh * 7560, 6000 -
((GlobalPhi(I + 1) + (InitialValue(I + 1) - GlobalPhi(I + 1)) * (1 - MultiFactor)) / InitialValue(1) * 4000 *
0.7 + 1000)), Cly
Next I
InoArea = 0
For I = 1 To NumNodes - 1
InoArea = InoArea + ((GlobalPhi(I) + (InitialValue(I) - GlobalPhi(I)) * (1 - MultiFactor)) + (GlobalPhi(I +
1) + (InitialValue(I + 1) - GlobalPhi(I + 1)) * (1 - MultiFactor))) / 2 * (Coord(I + 1) - Coord(I))
Next I
MigPercent = Format((IniArea - InoArea) / IniArea * 100, "fixed")
MigPpm = Format(MigPercent * ConcSubVial / 100, "fixed")
End Sub

Private Sub ComSimul_Click()
Open "c: \diffusions.dat" For Output As #1
Write #1, "Lumped"
Write #1, Val(NumNodesi.Text), Val(NumElei.Text), Val(NumMatlSetsi.Text)
Write #1, 0, 0, Val(NumKwnPhii.Text) 'Val(NumKwnSourcei.Text), Val(NumDerivBCi.Text),
For I = 1 To Val(NumMatlSetsi.Text):  Write #1, Val(txtDxi(I).Text), 0, 0, 1:  Next I
For I = 1 To Val(NumNodesi.Text)
Write #1, Val(Coordi(I).Text), Val(InitialValuei(I).Text)
Next I
For I = 1 To Val(NumElei.Text)
Write #1, Val(EleNodeDatai(I).Text), Val(EleNodeDataj(I).Text), Val(EleMatli(I).Text)
```

```
Next I
For I = 1 To Val(NumKwnPhii.Text):  Write #1, Val(SubKwnPhii(I).Text):  Next I
Write #1, 0.5, Val(DeltaTimei.Text)
Close #1
'Dimension the input data arrays
ReDim MatlValues(Val(NumMatlSetsi.Text), 4) As Single 'Material Values
ReDim Coord(Val(NumNodesi.Text)) As Single   'X Coordinate data
ReDim InitialValue(Val(NumNodesi.Text)) As Single     'Initial values of Phi
ReDim EleNodeData(Val(NumElei.Text), 2) As Integer   'Element node data
ReDim EleMatl(Val(NumElei.Text)) As Integer          'Element material index
ReDim SubKwnSource(1) As Integer
ReDim ValKwnSource(1) As Single
ReDim SubKwnDBC(1) As Integer  'Subscript of the derivative bdy condition
ReDim MValKwnDBC(1) As Single  M value for the derivative bdy condition
ReDim SValKwnDBC(1) As Single  'S value for the derivative bdy condition
ReDim SubKwnPhi(Val(NumKwnPhii.Text) + 1) As Integer  'Subscripts of the known Phi values.
Open "c: \diffusions.dat" For Input As #1
Input #1, cboType
Input #1, NumNodes, NumEle, NumMatlSets
Input #1, NumKwnSource, NumDerivBC, NumKwnPhi
'Input the basic data
Call InputMatlValues(NumMatlSets, MatlValues())
Call InputCoordData(NumNodes, Coord(), InitialValue())
Call InputNodeData(NumEle, EleNodeData(), EleMatl())
Call InputSourceVal(NumKwnSource, SubKwnSource(), ValKwnSource())
Call InputDerivBdyCd(NumDerivBC, SubKwnDBC(), MValKwnDBC(), SValKwnDBC())
Call InputKwnPhiData(NumKwnPhi, SubKwnPhi())
Input #1, Theta, DeltaTime
Close #1  'End of data input
'Dimension the element matrices
ReDim EFV(NumEleSub)              'Element force vector
ReDim ESM(NumEleSub, NumEleSub) As Single   'Element stiffness matrix
ReDim ECM(NumEleSub, NumEleSub) As Single   'Element capacitance matrix
ReDim EleSub(NumEleSub) As Integer  'Element displacement subscript values
'Dimension the arrays for the global system of equations
Dim NodalValuePerNode As Integer, NumNodalVal As Integer
NodalValuePerNode = 1:  NumNodalVal = NumNodes * NodalValuePerNode
Call CalBandWidth1D(NumEle, EleNodeData(), NodalValuePerNode, BandWidth)
ReDim GFV(NumNodalVal)            'Global force vector
ReDim GCM(NumNodalVal, BandWidth)  'Global capacitance matrix
ReDim GSM(NumNodalVal, BandWidth)  'Global stiffness matrix
ReDim GlobalPhi(NumNodalVal)       'New Phi values
ReDim ProductVector(NumNodalVal)   'Vector used in the solution process
'Build the system of equations element by element
Call ZeroGlobalMatrices(NumNodalVal, BandWidth, GFV(), GCM())
Call ZeroGlobalMatrices(NumNodalVal, BandWidth, GFV(), GCM())
For KK = 1 To NumEle
Call EleStiffMtx1DTime(cboType$, MatlValues!(), EleMatl%(), Coord!(), EleNodeData%(), EFV(),
ESM!(), ECM!(), KK)
Call EleSubscriptValues(KK%, EleNodeData%(), EleSub%())
Call BuildBandedTime(NumEleSub%, EleSub%(), EFV(), ECM!(), ESM!(), GFV(), GCM(), GSM())
Next KK
Call SourceSinkValue(NumKwnSource%, SubKwnSource%(), ValKwnSource!(), GFV())
Call DerivBdyCondition(NumDerivBC%, SubKwnDBC%(), MValKwnDBC!(), SValKwnDBC!(), GSM(),
GFV())
'Modify the [C] and [K] matrices to incorporate the known Phi values
```

```
'Convert [C] and [K] to [A] and [P]; Decompose [A]
Call DefineGlobalPhi(NumNodalVal%, InitialValue!(), GlobalPhi())
Call ModifyCandK(NumNodalVal%, BandWidth%, NumKwnPhi%, SubKwnPhi%(), GlobalPhi(), GFV(),
GSM(), GCM())
Call ConvertCandK(NumNodes%, BandWidth%, Theta, DeltaTime, GCM(), GSM(), GFV())
Call ConvertFvector(NumNodes%, Theta, DeltaTime, GFV())
Call DecompBandMatrix(NumNodalVal%, BandWidth%, GCM())
MultiFactor = 1 / (1 + PartitionC * WtPlasticS / DenFood / VoFoodS)
MaxMigConc = ConcSubVial: MaxCVial.Text = ConcSubVial
Clb = vbBlack: Clr = vbRed: Clbl = vbBlue: Cly = vbYellow: Clm = vbMagenta: Clc = vbCyan
Dim LineColor As Long
LineColor = RGB(175, 175, 250)
PictureSim.Cls
PictureSim.ScaleWidth = 12000: PictureSim.ScaleHeight = 6000
'x-gridlines
PictureSim.DrawWidth = 1
PictureSim.Line (0, 200)-(12000, 200), LineColor: PictureSim.Line (0, 1000)-(12000, 1000), LineColor
PictureSim.Line (0, 1800)-(12000, 1800), LineColor: PictureSim.Line (0, 2600)-(12000, 2600), LineColor
PictureSim.Line (0, 3400)-(12000, 3400), LineColor: PictureSim.Line (0, 4200)-(12000, 4200), LineColor
PictureSim.Line (0, 5000)-(12000, 5000), LineColor: PictureSim.Line (0, 5800)-(12000, 5800), LineColor
PictureSim.Line (0, 200)-(12000, 200), LineColor: PictureSim.Line (0, 600)-(12000, 600), LineColor
PictureSim.Line (0, 1400)-(12000, 1400), LineColor: PictureSim.Line (0, 2200)-(12000, 2200), LineColor
PictureSim.Line (0, 3000)-(12000, 3000), LineColor: PictureSim.Line (0, 3800)-(12000, 3800), LineColor
PictureSim.Line (0, 4600)-(12000, 4600), LineColor: PictureSim.Line (0, 5400)-(12000, 5400), LineColor
'y-gridlines
PictureSim.Line (50, 0)-(50, 6000), LineColor: PictureSim.Line (1750, 0)-(1750, 6000), LineColor
PictureSim.Line (3450, 0)-(3450, 6000), LineColor: PictureSim.Line (5150, 0)-(5150, 6000), LineColor
PictureSim.Line (6850, 0)-(6850, 6000), LineColor: PictureSim.Line (8550, 0)-(8550, 6000), LineColor
PictureSim.Line (10250, 0)-(10250, 6000), LineColor: PictureSim.Line (11950, 0)-(11950, 6000),
LineColor
PictureSim.Line (50, 0)-(50, 6000), LineColor: PictureSim.Line (900, 0)-(900, 6000), LineColor
PictureSim.Line (2600, 0)-(2600, 6000), LineColor: PictureSim.Line (4300, 0)-(4300, 6000), LineColor
PictureSim.Line (6000, 0)-(6000, 6000), LineColor: PictureSim.Line (7700, 0)-(7700, 6000), LineColor
PictureSim.Line (9400, 0)-(9400, 6000), LineColor: PictureSim.Line (11100, 0)-(11100, 6000), LineColor
PictureSim.DrawWidth = 2 'draw X-Y axis and boundary
PictureSim.Line (1750, 6000 - 1000)-(11100, 6000 - 1000), Clb    'X axis
PictureSim.Line (1750, 6000 - 1000)-(1750, 6000 - 5400), Clb    'Y axis
PictureSim.CurrentX = 1700: PictureSim.CurrentY = 5100: PictureSim.Print "0"
PictureSim.CurrentX = 1400: PictureSim.CurrentY = 4800: PictureSim.Print "0"
PictureSim.DrawWidth = 2
PictureSim.Line (1750, 1000)-(1850, 1000): PictureSim.Line (1750, 1800)-(1850, 1800)
PictureSim.Line (1750, 2600)-(1850, 2600): PictureSim.Line (1750, 3400)-(1850, 3400)
PictureSim.Line (1750, 4200)-(1850, 4200)
PictureSim.CurrentX = 1000: PictureSim.CurrentY = 900: PictureSim.Print "100%"
PictureSim.CurrentX = 1150: PictureSim.CurrentY = 1700: PictureSim.Print "80%"
PictureSim.CurrentX = 1150: PictureSim.CurrentY = 2500: PictureSim.Print "60%"
PictureSim.CurrentX = 1150: PictureSim.CurrentY = 3300: PictureSim.Print "40%"
PictureSim.CurrentX = 1150: PictureSim.CurrentY = 4100: PictureSim.Print "20%"
PictureSim.CurrentX = 400: PictureSim.CurrentY = 500: PictureSim.Print "Mt/Me"
If AnyTimeT(Val(NumMigData.Text)) <= 5 Then          MaxXax = 5
ElseIf AnyTimeT(Val(NumMigData.Text)) <= 6 Then      MaxXax = 6
ElseIf AnyTimeT(Val(NumMigData.Text)) <= 7 Then      MaxXax = 7
ElseIf AnyTimeT(Val(NumMigData.Text)) <= 8 Then      MaxXax = 8
ElseIf AnyTimeT(Val(NumMigData.Text)) <= 9 Then      MaxXax = 9
ElseIf AnyTimeT(Val(NumMigData.Text)) <= 10 Then     MaxXax = 10
ElseIf AnyTimeT(Val(NumMigData.Text)) <= 12 Then     MaxXax = 12
```

```
ElseIf AnyTimeT(Val(NumMigData.Text)) <= 15 Then     MaxXax = 15
ElseIf AnyTimeT(Val(NumMigData.Text)) <= 18 Then     MaxXax = 18
ElseIf AnyTimeT(Val(NumMigData.Text)) <= 20 Then     MaxXax = 20
ElseIf AnyTimeT(Val(NumMigData.Text)) <= 25 Then     MaxXax = 25
ElseIf AnyTimeT(Val(NumMigData.Text)) <= 30 Then     MaxXax = 30
ElseIf AnyTimeT(Val(NumMigData.Text)) <= 40 Then     MaxXax = 40
ElseIf AnyTimeT(Val(NumMigData.Text)) <= 50 Then     MaxXax = 50
Else      MaxXax = 100
End If
TimeCalcSim.Text = MaxXax ^ 2:  PictureSim.DrawWidth = 2
PictureSim.Line (10250, 4930)-(10250, 5000):  PictureSim.Line (8550, 4930)-(8550, 5000)
PictureSim.Line (6850, 4930)-(6850, 5000):  PictureSim.Line (5150, 4930)-(5150, 5000)
PictureSim.Line (3450, 4930)-(3450, 5000):
PictureSim.CurrentX = 10000:  PictureSim.CurrentY = 5100:  PictureSim.Print Format(MaxXax, "fixed")
PictureSim.CurrentX = 8300:  PictureSim.CurrentY = 5100
PictureSim.Print Format(MaxXax * 0.8, "fixed")
PictureSim.CurrentX = 6600:  PictureSim.CurrentY = 5100
PictureSim.Print Format(MaxXax * 0.6, "fixed")
PictureSim.CurrentX = 4900:  PictureSim.CurrentY = 5100
PictureSim.Print Format(MaxXax * 0.4, "fixed")
PictureSim.CurrentX = 3200:  PictureSim.CurrentY = 5100
PictureSim.Print Format(MaxXax * 0.2, "fixed")
PictureSim.CurrentX = 10650:  PictureSim.CurrentY = 5100:  PictureSim.Print " hrs½"
PictureSim.CurrentX = 10650:  PictureSim.CurrentY = 5500:  PictureSim.Print " days"
PictureSim.CurrentX = 10000:  PictureSim.CurrentY = 5500
PictureSim.Print Format(MaxXax ^ 2 / 24, "fixed")
PictureSim.CurrentX = 8300:  PictureSim.CurrentY = 5500
PictureSim.Print Format((MaxXax * 0.8) ^ 2 / 24, "fixed")
PictureSim.CurrentX = 6600:  PictureSim.CurrentY = 5500
PictureSim.Print Format((MaxXax * 0.6) ^ 2 / 24, "fixed")
PictureSim.CurrentX = 4900:  PictureSim.CurrentY = 5500
PictureSim.Print Format((MaxXax * 0.4) ^ 2 / 24, "fixed")
PictureSim.CurrentX = 3200:  PictureSim.CurrentY = 5500
PictureSim.Print Format((MaxXax * 0.2) ^ 2 / 24, "fixed")
'Test Conc. at each time interval
PictureSim.DrawWidth = 6
For J = 1 To NumMigData
PictureSim.Line (1750 + AnyTimeT(J) / MaxXax * 8500, 5000 - AnyConc(J) / MaxMigConc * 4000)-
(1800 + AnyTimeT(J) / MaxXax * 8500, 5000 - AnyConc(J) / MaxMigConc * 4000), Clbl
Next J
'Do the solution in time durations.
Call DefineGlobalPhi(NumNodalVal%, InitialValue!(), GlobalPhi())
KJ = 1
Timet = 0
NumTimeSteps = (MaxXax ^ 2) * 3600 / DeltaTime
For KJ = 1 To NumTimeSteps Step 1
Timet = Timet + DeltaTime
Call MultpyBandMatrix(NumNodalVal%, BandWidth%, GSM(), GlobalPhi(), ProductVector())
For I = 1 To NumNodalVal%
ProductVector(I) = ProductVector(I) + GFV(I)
Next I
Call SolveBandMatrix(NumNodalVal%, BandWidth%, GlobalPhi(), ProductVector(), GCM())
Call OutputScreen(KJ, NumNodalVal%, Timet, GlobalPhi())
'Area-initial under the line
IniArea = 0
For I = 1 To NumNodes - 1
```

```
IniArea = IniArea + (InitialValue(I) + InitialValue(I + 1)) / 2 * (Coord(I + 1) - Coord(I))
Next I
InoArea = 0
For I = 1 To NumNodes - 1
InoArea = InoArea + ((GlobalPhi(I) + (InitialValue(I) - GlobalPhi(I)) * (1 - MultiFactor)) + (GlobalPhi(I +
1) + (InitialValue(I + 1) - GlobalPhi(I + 1)) * (1 - MultiFactor))) / 2 * (Coord(I + 1) - Coord(I))
MigPercent = (IniArea - InoArea) / IniArea * 100
PictureSim.DrawWidth = 3
PictureSim.Line (1750 + ((Timet / 3600) ^ 0.5) / MaxXax * 8500, 5000 - MigPercent / 100 * 4000)-(1780
+ ((Timet / 3600) ^ 0.5) / MaxXax * 8500, 5000 - MigPercent / 100 * 4000), Clr
Next KJ
End Sub

Private Sub cmdSimOnly_Click()
Open "c: \diffusiono.dat" For Output As #1
Write #1, "Lumped"
Write #1, Val(NumNodesi.Text), Val(NumElei.Text), Val(NumMatlSetsi.Text)
Write #1, 0, 0, Val(NumKwnPhii.Text) 'Val(NumKwnSourcei.Text), Val(NumDerivBCi.Text),
For I = 1 To Val(NumMatlSetsi.Text): Write #1, Val(txtDxi(I).Text), 0, 0, 1: Next I
For I = 1 To Val(NumNodesi.Text)
Write #1, Val(Coordi(I).Text), Val(InitialValuei(I).Text)
Next I
For I = 1 To Val(NumElei.Text)
Write #1, Val(EleNodeDatai(I).Text), Val(EleNodeDataj(I).Text), Val(EleMatli(I).Text)
Next I
For I = 1 To Val(NumKwnPhii.Text): Write #1, Val(SubKwnPhii(I).Text): Next I
Write #1, 0.5, Val(DeltaTimei.Text)
Close #1
'Dimension the input data arrays
ReDim MatlValues(Val(NumMatlSetsi.Text), 4) As Single 'Material Values
ReDim Coord(Val(NumNodesi.Text)) As Single   'X Coordinate data
ReDim InitialValue(Val(NumNodesi.Text)) As Single    'Initial values of Phi
ReDim EleNodeData(Val(NumElei.Text), 2) As Integer   'Element node data
ReDim EleMatl(Val(NumElei.Text)) As Integer          'Element material index
ReDim SubKwnSource(1) As Integer
ReDim ValKwnSource(1) As Single
ReDim SubKwnDBC(1) As Integer 'Subscript of the derivative bdy condition
ReDim MValKwnDBC(1) As Single  'M value for the derivative bdy condition
ReDim SValKwnDBC(1) As Single  'S value for the derivative bdy condition
ReDim SubKwnPhi(Val(NumKwnPhii.Text) + 1) As Integer   'Subscripts of the known Phi values.
Open "c: \diffusiono.dat" For Input As #1
Input #1, cboType
Input #1, NumNodes, NumEle, NumMatlSets
Input #1, NumKwnSource, NumDerivBC, NumKwnPhi
'Input the basic data
Call InputMatlValues(NumMatlSets, MatlValues())
Call InputCoordData(NumNodes, Coord(), InitialValue())
Call InputNodeData(NumEle, EleNodeData(), EleMatl())
Call InputSourceVal(NumKwnSource, SubKwnSource(), ValKwnSource())
Call InputDerivBdyCd(NumDerivBC, SubKwnDBC(), MValKwnDBC(), SValKwnDBC())
Call InputKwnPhiData(NumKwnPhi, SubKwnPhi())
Input #1, Theta, DeltaTime
Close #1  'End of data input
'Dimension the element matrices
ReDim EFV(NumEleSub)             'Element force vector
ReDim ESM(NumEleSub, NumEleSub) As Single   'Element stiffness matrix
```

```
ReDim ECM(NumEleSub, NumEleSub) As Single   'Element capacitance matrix
ReDim EleSub(NumEleSub) As Integer 'Element displacement subscript values
'Dimension the arrays for the global system of equations
Dim NodalValuePerNode As Integer, NumNodalVal As Integer
NodalValuePerNode = 1:  NumNodalVal = NumNodes * NodalValuePerNode
Call CalBandWidth1D(NumEle, EleNodeData(), NodalValuePerNode, BandWidth)
ReDim GFV(NumNodalVal)          'Global force vector
ReDim GCM(NumNodalVal, BandWidth)  'Global capacitance matrix
ReDim GSM(NumNodalVal, BandWidth)  'Global stiffness matrix
ReDim GlobalPhi(NumNodalVal)       'New Phi values
ReDim ProductVector(NumNodalVal)   'Vector used in the solution process
'Build the system of equations element by element
Call ZeroGlobalMatrices(NumNodalVal, BandWidth, GFV(), GCM())
Call ZeroGlobalMatrices(NumNodalVal, BandWidth, GFV(), GCM())
For KK = 1 To NumEle
Call EleStiffMtx1DTime(cboType$, MatlValues!(), EleMatl%(), Coord!(), EleNodeData%(), EFV(),
ESM!(), ECM!(), KK)
Call EleSubscriptValues(KK%, EleNodeData%(), EleSub%())
Call BuildBandedTime(NumEleSub%, EleSub%(), EFV(), ECM!(), ESM!(), GFV(), GCM(), GSM())
Next KK
Call SourceSinkValue(NumKwnSource%, SubKwnSource%(), ValKwnSource!(), GFV())
Call DerivBdyCondition(NumDerivBC%, SubKwnDBC%(), MValKwnDBC!(), SValKwnDBC!(), GSM(),
GFV())
'Modify the [C] and [K] matrices to incorporate the known Phi values
'Convert [C] and [K] to [A] and [P]; Decompose [A]
Call DefineGlobalPhi(NumNodalVal%, InitialValue!(), GlobalPhi())
Call ModifyCandK(NumNodalVal%, BandWidth%, NumKwnPhi%, SubKwnPhi%(), GlobalPhi(), GFV(),
GSM(), GCM())
Call ConvertCandK(NumNodes%, BandWidth%, Theta, DeltaTime, GCM(), GSM(), GFV())
Call ConvertFvector(NumNodes%, Theta, DeltaTime, GFV())
Call DecompBandMatrix(NumNodalVal%, BandWidth%, GCM())
MultiFactor = 1 / (1 + PartitionC * WtPlasticS / DenFood / VoFoodS)
MaxCVial.Text = ConcSubVial
Clb = vbBlack:  Clr = vbRed:  Clbl = vbBlue:  Cly = vbYellow:  Clm = vbMagenta:  Clc = vbCyan
Dim LineColor As Long
LineColor = RGB(175, 175, 250)
PictureSim.Cls
PictureSim.ScaleWidth = 12000: PictureSim.ScaleHeight = 6000
'x-gridlines
PictureSim.DrawWidth = 1
PictureSim.Line (0, 200)-(12000, 200), LineColor: PictureSim.Line (0, 1000)-(12000, 1000), LineColor
PictureSim.Line (0, 1800)-(12000, 1800), LineColor: PictureSim.Line (0, 2600)-(12000, 2600), LineColor
PictureSim.Line (0, 3400)-(12000, 3400), LineColor: PictureSim.Line (0, 4200)-(12000, 4200), LineColor
PictureSim.Line (0, 5000)-(12000, 5000), LineColor: PictureSim.Line (0, 5800)-(12000, 5800), LineColor
PictureSim.Line (0, 200)-(12000, 200), LineColor: PictureSim.Line (0, 600)-(12000, 600), LineColor
PictureSim.Line (0, 1400)-(12000, 1400), LineColor: PictureSim.Line (0, 2200)-(12000, 2200), LineColor
PictureSim.Line (0, 3000)-(12000, 3000), LineColor: PictureSim.Line (0, 3800)-(12000, 3800), LineColor
PictureSim.Line (0, 4600)-(12000, 4600), LineColor: PictureSim.Line (0, 5400)-(12000, 5400), LineColor
'y-gridlines
PictureSim.Line (50, 0)-(50, 6000), LineColor: PictureSim.Line (1750, 0)-(1750, 6000), LineColor
PictureSim.Line (3450, 0)-(3450, 6000), LineColor: PictureSim.Line (5150, 0)-(5150, 6000), LineColor
PictureSim.Line (6850, 0)-(6850, 6000), LineColor: PictureSim.Line (8550, 0)-(8550, 6000), LineColor
PictureSim.Line (10250, 0)-(10250, 6000), LineColor: PictureSim.Line (11950, 0)-(11950, 6000),
LineColor
PictureSim.Line (50, 0)-(50, 6000), LineColor: PictureSim.Line (900, 0)-(900, 6000), LineColor
PictureSim.Line (2600, 0)-(2600, 6000), LineColor: PictureSim.Line (4300, 0)-(4300, 6000), LineColor
```

```
PictureSim.Line (6000, 0)-(6000, 6000), LineColor: PictureSim.Line (7700, 0)-(7700, 6000), LineColor
PictureSim.Line (9400, 0)-(9400, 6000), LineColor: PictureSim.Line (11100, 0)-(11100, 6000), LineColor
PictureSim.DrawWidth = 2 'draw X-Y axis and boundary
PictureSim.Line (1750, 6000 - 1000)-(11100, 6000 - 1000), Clb    'X axis
PictureSim.Line (1750, 6000 - 1000)-(1750, 6000 - 5400), Clb    'Y axis
PictureSim.CurrentX = 1700:  PictureSim.CurrentY = 5100: PictureSim.Print "0"
PictureSim.CurrentX = 1400: PictureSim.CurrentY = 4800: PictureSim.Print "0"
PictureSim.DrawWidth = 2
PictureSim.Line (1750, 1000)-(1850, 1000): PictureSim.Line (1750, 1800)-(1850, 1800)
PictureSim.Line (1750, 2600)-(1850, 2600): PictureSim.Line (1750, 3400)-(1850, 3400)
PictureSim.Line (1750, 4200)-(1850, 4200)
PictureSim.CurrentX = 1000: PictureSim.CurrentY = 900: PictureSim.Print "100%"
PictureSim.CurrentX = 1150: PictureSim.CurrentY = 1700: PictureSim.Print "80%"
PictureSim.CurrentX = 1150: PictureSim.CurrentY = 2500: PictureSim.Print "60%"
PictureSim.CurrentX = 1150: PictureSim.CurrentY = 3300: PictureSim.Print "40%"
PictureSim.CurrentX = 1150: PictureSim.CurrentY = 4100: PictureSim.Print "20%"
PictureSim.CurrentX = 400: PictureSim.CurrentY = 500: PictureSim.Print "Mt/Me"
TimeSqrt = TimeSimOnly ^ 0.5
If TimeSqrt <= 5 Then       MaxXax = 5
ElseIf TimeSqrt <= 6 Then       MaxXax = 6 :  ElseIf TimeSqrt <= 7 Then       MaxXax = 7
ElseIf TimeSqrt <= 8 Then       MaxXax = 8 :  ElseIf TimeSqrt <= 9 Then       MaxXax = 9
ElseIf TimeSqrt <= 10 Then      MaxXax = 10: ElseIf TimeSqrt <= 12 Then      MaxXax = 12
ElseIf TimeSqrt <= 15 Then      MaxXax = 15: ElseIf TimeSqrt <= 18 Then      MaxXax = 18
ElseIf TimeSqrt <= 20 Then      MaxXax = 20: ElseIf TimeSqrt <= 25 Then      MaxXax = 25
ElseIf TimeSqrt <= 30 Then      MaxXax = 30: ElseIf TimeSqrt <= 40 Then      MaxXax = 40
ElseIf TimeSqrt <= 50 Then      MaxXax = 50: Else                            MaxXax = 100
End If
PictureSim.DrawWidth = 2
PictureSim.Line (10250, 4930)-(10250, 5000): PictureSim.Line (8550, 4930)-(8550, 5000)
PictureSim.Line (6850, 4930)-(6850, 5000): PictureSim.Line (5150, 4930)-(5150, 5000)
PictureSim.Line (3450, 4930)-(3450, 5000)
PictureSim.CurrentX = 10000: PictureSim.CurrentY = 5100
PictureSim.Print Format(MaxXax, "fixed")
PictureSim.CurrentX = 8300: PictureSim.CurrentY = 5100
PictureSim.Print Format(MaxXax * 0.8, "fixed")
PictureSim.CurrentX = 6600: PictureSim.CurrentY = 5100
PictureSim.Print Format(MaxXax * 0.6, "fixed")
PictureSim.CurrentX = 4900: PictureSim.CurrentY = 5100
PictureSim.Print Format(MaxXax * 0.4, "fixed")
PictureSim.CurrentX = 3200: PictureSim.CurrentY = 5100
PictureSim.Print Format(MaxXax * 0.2, "fixed")
PictureSim.CurrentX = 10650: PictureSim.CurrentY = 5100: PictureSim.Print " hrs½"
PictureSim.CurrentX = 10650: PictureSim.CurrentY = 5500: PictureSim.Print " days"
PictureSim.CurrentX = 10000: PictureSim.CurrentY = 5500
PictureSim.Print Format(MaxXax ^ 2 / 24, "fixed")
PictureSim.CurrentX = 8300: PictureSim.CurrentY = 5500
PictureSim.Print Format((MaxXax * 0.8) ^ 2 / 24, "fixed")
PictureSim.CurrentX = 6600: PictureSim.CurrentY = 5500
PictureSim.Print Format((MaxXax * 0.6) ^ 2 / 24, "fixed")
PictureSim.CurrentX = 4900: PictureSim.CurrentY = 5500
PictureSim.Print Format((MaxXax * 0.4) ^ 2 / 24, "fixed")
PictureSim.CurrentX = 3200: PictureSim.CurrentY = 5500
PictureSim.Print Format((MaxXax * 0.2) ^ 2 / 24, "fixed")
'Do the solution in time durations.
Call DefineGlobalPhi(NumNodalVal%, InitialValue!(), GlobalPhi())
KJ = 1
```

```
Timet = 0
NumTimeSteps = Val(TimeSimOnly.Text) * 3600 / DeltaTime
For KJ = 1 To NumTimeSteps Step 1
Timet = Timet + DeltaTime
Call MultpyBandMatrix(NumNodalVal%, BandWidth%, GSM(), GlobalPhi(), ProductVector())
For I = 1 To NumNodalVal%
ProductVector(I) = ProductVector(I) + GFV(I)
Next I
Call SolveBandMatrix(NumNodalVal%, BandWidth%, GlobalPhi(), ProductVector(), GCM())
Call OutputScreen(KJ, NumNodalVal%, Timet, GlobalPhi())
'Area-initial under the line
IniArea = 0
For I = 1 To NumNodes - 1
IniArea = IniArea + (InitialValue(I) + InitialValue(I + 1)) / 2 * (Coord(I + 1) - Coord(I))
Next I
InoArea = 0
For I = 1 To NumNodes - 1
InoArea = InoArea + ((GlobalPhi(I) + (InitialValue(I) - GlobalPhi(I)) * (1 - MultiFactor)) + (GlobalPhi(I +
1) + (InitialValue(I + 1) - GlobalPhi(I + 1)) * (1 - MultiFactor))) / 2 * (Coord(I + 1) - Coord(I))
Next I
MigPercent = (IniArea - InoArea) / IniArea * 100
PictureSim.DrawWidth = 5
PictureSim.Line (1750 + ((Timet / 3600) ^ 0.5) / MaxXax * 8500, 5000 - MigPercent / 100 * 4000)-(1770
+ ((Timet / 3600) ^ 0.5) / MaxXax * 8500, 5000 - MigPercent / 100 * 4000), Clr
Next KJ
End Sub

Sub OutputScreen(KK, NumNodalVal%, Timet, GlobalPhi())
For I = 1 To NumNodalVal
Label2(I).Caption = "Node No.[" & I & "]"
Next I
For J = 1 To NumNodalVal
Text1(J).Text = GlobalPhi(J) + (InitialValue(J) - GlobalPhi(J)) * (1 - MultiFactor)
Next J
End Sub

Private Sub cmdSaveSim_Click()
Open "c: \diffusiono.dat" For Output As #1
Write #1, "Lumped"
Write #1, Val(NumNodesi.Text), Val(NumElei.Text), Val(NumMatlSetsi.Text)
Write #1, 0, 0, Val(NumKwnPhii.Text) 'Val(NumKwnSourcei.Text), Val(NumDerivBCi.Text),
For I = 1 To Val(NumMatlSetsi.Text): Write #1, Val(txtDxi(I).Text), 0, 0, 1: Next I
For I = 1 To Val(NumNodesi.Text)
Write #1, Val(Coordi(I).Text), Val(InitialValuei(I).Text)
Next I
For I = 1 To Val(NumElei.Text)
Write #1, Val(EleNodeDatai(I).Text), Val(EleNodeDataj(I).Text), Val(EleMatli(I).Text)
Next I
For I = 1 To Val(NumKwnPhii.Text): Write #1, Val(SubKwnPhii(I).Text): Next I
Write #1, 0.5, Val(DeltaTimei.Text)
Close #1
'Dimension the input data arrays
ReDim MatlValues(Val(NumMatlSetsi.Text), 4) As Single 'Material Values
ReDim Coord(Val(NumNodesi.Text)) As Single   'X Coordinate data
ReDim InitialValue(Val(NumNodesi.Text)) As Single     'Initial values of Phi
ReDim EleNodeData(Val(NumElei.Text), 2) As Integer   'Element node data
```

```
ReDim EleMatl(Val(NumElei.Text)) As Integer          'Element material index
ReDim SubKwnSource(1) As Integer
ReDim ValKwnSource(1) As Single
ReDim SubKwnDBC(1) As Integer 'Subscript of the derivative bdy condition
ReDim MValKwnDBC(1) As Single  'M value for the derivative bdy condition
ReDim SValKwnDBC(1) As Single  'S value for the derivative bdy condition
ReDim SubKwnPhi(Val(NumKwnPhii.Text) + 1) As Integer   'Subscripts of the known Phi values.
Open "c: \diffusiono.dat" For Input As #1
Input #1, cboType
Input #1, NumNodes, NumEle, NumMatlSets
Input #1, NumKwnSource, NumDerivBC, NumKwnPhi
'Input the basic data
Call InputMatlValues(NumMatlSets, MatlValues())
Call InputCoordData(NumNodes, Coord(), InitialValue())
Call InputNodeData(NumEle, EleNodeData(), EleMatl())
Call InputSourceVal(NumKwnSource, SubKwnSource(), ValKwnSource())
Call InputDerivBdyCd(NumDerivBC, SubKwnDBC(), MValKwnDBC(), SValKwnDBC())
Call InputKwnPhiData(NumKwnPhi, SubKwnPhi())
Input #1, Theta, DeltaTime
Close #1  'End of data input
'Dimension the element matrices
ReDim EFV(NumEleSub)              'Element force vector
ReDim ESM(NumEleSub, NumEleSub) As Single   'Element stiffness matrix
ReDim ECM(NumEleSub, NumEleSub) As Single   'Element capacitance matrix
ReDim EleSub(NumEleSub) As Integer  'Element displacement subscript values
'Dimension the arrays for the global system of equations
Dim NodalValuePerNode As Integer, NumNodalVal As Integer
NodalValuePerNode = 1:  NumNodalVal = NumNodes * NodalValuePerNode
Call CalBandWidth1D(NumEle, EleNodeData(), NodalValuePerNode, BandWidth)
ReDim GFV(NumNodalVal)            'Global force vector
ReDim GCM(NumNodalVal, BandWidth)  'Global capacitance matrix
ReDim GSM(NumNodalVal, BandWidth)  'Global stiffness matrix
ReDim GlobalPhi(NumNodalVal)       'New Phi values
ReDim ProductVector(NumNodalVal)   'Vector used in the solution process
'Build the system of equations element by element
Call ZeroGlobalMatrices(NumNodalVal, BandWidth, GFV(), GCM())
Call ZeroGlobalMatrices(NumNodalVal, BandWidth, GFV(), GCM())
For KK = 1 To NumEle
Call EleStiffMtx1DTime(cboType$, MatlValues!(), EleMatl%(), Coord!(), EleNodeData%(), EFV(),
ESM!(), ECM!(), KK)
Call EleSubscriptValues(KK%, EleNodeData%(), EleSub%())
Call BuildBandedTime(NumEleSub%, EleSub%(), EFV(), ECM!(), ESM!(), GFV(), GCM(), GSM())
Next KK
Call SourceSinkValue(NumKwnSource%, SubKwnSource%(), ValKwnSource!(), GFV())
Call DerivBdyCondition(NumDerivBC%, SubKwnDBC%(), MValKwnDBC!(), SValKwnDBC!(), GSM(),
GFV())
'Modify the [C] and [K] matrices to incorporate the known Phi values
'Convert [C] and [K] to [A] and [P]; Decompose [A]
Call DefineGlobalPhi(NumNodalVal%, InitialValue!(), GlobalPhi())
Call ModifyCandK(NumNodalVal%, BandWidth%, NumKwnPhi%, SubKwnPhi%(), GlobalPhi(), GFV(),
GSM(), GCM())
Call ConvertCandK(NumNodes%, BandWidth%, Theta, DeltaTime, GCM(), GSM(), GFV())
Call ConvertFvector(NumNodes%, Theta, DeltaTime, GFV())
Call DecompBandMatrix(NumNodalVal%, BandWidth%, GCM())
MultiFactor = 1 / (1 + PartitionC * WtPlasticS / DenFood / VoFoodS)
MaxCVial.Text = ConcSubVial
```

```
TimeSqrt = TimeSimOnly ^ 0.5
If TimeSqrt <= 5 Then       MaxXax = 5
ElseIf TimeSqrt <= 6 Then       MaxXax = 6  : ElseIf TimeSqrt <= 7 Then       MaxXax = 7
ElseIf TimeSqrt <= 8 Then       MaxXax = 8  : ElseIf TimeSqrt <= 9 Then       MaxXax = 9
ElseIf TimeSqrt <= 10 Then      MaxXax = 10: ElseIf TimeSqrt <= 12 Then      MaxXax = 12
ElseIf TimeSqrt <= 15 Then      MaxXax = 15: ElseIf TimeSqrt <= 18 Then      MaxXax = 18
ElseIf TimeSqrt <= 20 Then      MaxXax = 20: ElseIf TimeSqrt <= 25 Then      MaxXax = 25
ElseIf TimeSqrt <= 30 Then      MaxXax = 30: ElseIf TimeSqrt <= 40 Then      MaxXax = 40
ElseIf TimeSqrt <= 50 Then      MaxXax = 50: Else                            MaxXax = 100
End If
'Do the solution in time durations.
Call DefineGlobalPhi(NumNodalVal%, InitialValue!(), GlobalPhi())
KJ = 1
Timet = 0
NumTimeSteps = Val(TimeSimOnly.Text) * 3600 / DeltaTime
MigPercent = 0
Open "c: \Diffussim.dat" For Output As #4
Write #4, NumTimeSteps
Write #4, Timet, MigPercentG
Close #4
For KJ = 1 To NumTimeSteps Step 1
Timet = Timet + DeltaTime
Call MultpyBandMatrix(NumNodalVal%, BandWidth%, GSM(), GlobalPhi(), ProductVector())
For I = 1 To NumNodalVal%
ProductVector(I) = ProductVector(I) + GFV(I)
Next I
Call SolveBandMatrix(NumNodalVal%, BandWidth%, GlobalPhi(), ProductVector(), GCM())
Call OutputScreen(KJ, NumNodalVal%, Timet, GlobalPhi())
'Area-initial under the line
IniArea = 0
For I = 1 To NumNodes - 1
IniArea = IniArea + (InitialValue(I) + InitialValue(I + 1)) / 2 * (Coord(I + 1) - Coord(I))
Next I
InoArea = 0
For I = 1 To NumNodes - 1
InoArea = InoArea + ((GlobalPhi(I) + (InitialValue(I) - GlobalPhi(I)) * (1 - MultiFactor)) + (GlobalPhi(I +
1) + (InitialValue(I + 1) - GlobalPhi(I + 1)) * (1 - MultiFactor))) / 2 * (Coord(I + 1) - Coord(I))
Next I
MigPercentG = (IniArea - InoArea) / IniArea * 100
Open "c: \Diffussim.dat" For Append As #4
Write #4, Timet, MigPercentG
Close #4
Next KJ
End Sub

Private Sub cmdPreSim_Click()
Open "c: \Diffussim.dat" For Input As #4
Input #4, GrpNumTimeS
ReDim GrpTimet(GrpNumTimeS + 1) As Single
ReDim GrpMigPer(GrpNumTimeS + 1) As Single
For I = 1 To GrpNumTimeS
Input #4, GrpTimet(I), GrpMigPer(I)
Next I
Close #4
Clb = vbBlack: Clr = vbRed: Clbl = vbBlue: Cly = vbYellow: Clm = vbMagenta: Clc = vbCyan
PictureSim.CurrentX = 9750: PictureSim.CurrentY = 4500: PictureSim.Print "       "
```

```
TimeSqrt = TimeSimOnly ^ 0.5
If TimeSqrt <= 5 Then       MaxXax = 5
ElseIf TimeSqrt <= 6 Then       MaxXax = 6  : ElseIf TimeSqrt <= 7 Then       MaxXax = 7
ElseIf TimeSqrt <= 8 Then       MaxXax = 8  : ElseIf TimeSqrt <= 9 Then       MaxXax = 9
ElseIf TimeSqrt <= 10 Then      MaxXax = 10: ElseIf TimeSqrt <= 12 Then      MaxXax = 12
ElseIf TimeSqrt <= 15 Then      MaxXax = 15: ElseIf TimeSqrt <= 18 Then      MaxXax = 18
ElseIf TimeSqrt <= 20 Then      MaxXax = 20: ElseIf TimeSqrt <= 25 Then      MaxXax = 25
ElseIf TimeSqrt <= 30 Then      MaxXax = 30: ElseIf TimeSqrt <= 40 Then      MaxXax = 40
ElseIf TimeSqrt <= 50 Then      MaxXax = 50: Else                            MaxXax = 100
End If
Dim PrevColor As Long
PrevColor = RGB(0, 120, 0)
PictureSim.DrawWidth = 3
For J = 1 To GrpNumTimeS
PictureSim.Line (1750 + ((GrpTimet(J) / 3600) ^ 0.5) / MaxXax * 8500, 5000 - GrpMigPer(J) / 100 *
4000)-(1770 + ((GrpTimet(J) / 3600) ^ 0.5) / MaxXax * 8500, 5000 - GrpMigPer(J) / 100 * 4000),
PrevColor
Next J
End Sub

Private Sub cmdClear_Click()
Dim LineColor As Long
LineColor = RGB(175, 175, 250)
PictureSim.Cls
PictureSim.ScaleWidth = 12000:  PictureSim.ScaleHeight = 6000
PictureSim.DrawWidth = 1
PictureSim.Line (0, 200)-(12000, 200), LineColor: PictureSim.Line (0, 1000)-(12000, 1000), LineColor
PictureSim.Line (0, 1800)-(12000, 1800), LineColor: PictureSim.Line (0, 2600)-(12000, 2600), LineColor
PictureSim.Line (0, 3400)-(12000, 3400), LineColor: PictureSim.Line (0, 4200)-(12000, 4200), LineColor
PictureSim.Line (0, 5000)-(12000, 5000), LineColor: PictureSim.Line (0, 5800)-(12000, 5800), LineColor
PictureSim.Line (0, 200)-(12000, 200), LineColor: PictureSim.Line (0, 600)-(12000, 600), LineColor
PictureSim.Line (0, 1400)-(12000, 1400), LineColor: PictureSim.Line (0, 2200)-(12000, 2200), LineColor
PictureSim.Line (0, 3000)-(12000, 3000), LineColor: PictureSim.Line (0, 3800)-(12000, 3800), LineColor
PictureSim.Line (0, 4600)-(12000, 4600), LineColor: PictureSim.Line (0, 5400)-(12000, 5400), LineColor
'y-gridlines
PictureSim.Line (50, 0)-(50, 6000), LineColor: PictureSim.Line (1750, 0)-(1750, 6000), LineColor
PictureSim.Line (3450, 0)-(3450, 6000), LineColor: PictureSim.Line (5150, 0)-(5150, 6000), LineColor
PictureSim.Line (6850, 0)-(6850, 6000), LineColor: PictureSim.Line (8550, 0)-(8550, 6000), LineColor
PictureSim.Line (10250, 0)-(10250, 6000), LineColor: PictureSim.Line (11950, 0)-(11950, 6000),
LineColor
PictureSim.Line (50, 0)-(50, 6000), LineColor: PictureSim.Line (900, 0)-(900, 6000), LineColor
PictureSim.Line (2600, 0)-(2600, 6000), LineColor: PictureSim.Line (4300, 0)-(4300, 6000), LineColor
PictureSim.Line (6000, 0)-(6000, 6000), LineColor: PictureSim.Line (7700, 0)-(7700, 6000), LineColor
PictureSim.Line (9400, 0)-(9400, 6000), LineColor: PictureSim.Line (11100, 0)-(11100, 6000), LineColor
End Sub

Private Sub PictureSim_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
txtXTime.Text = Format(((X - 1750) * MaxXax / 8500) ^ 2, ".#")
txtYPercent.Text = Format((5000 - Y) / 40, ".#")
End Sub
```

Module

"dimModule" Code

```
DefInt I-N
'Dimension the input data arrays
Public I As Integer, J As Integer, K As Integer, KJ As Integer,
Public II As Integer, KK As Integer, MM As Integer, JJ As Integer
Public cboTitle As String, cboTitlei As String, cboType As String, cboTypei As String
Public NumMatlSets As Integer, NumMatlSetsi As Integer, NumNodes As Integer, NumNodesi As Integer
Public NumEle As Integer, NumElei As Integer, NumKwnSource As Integer, NumKwnSoucei As Integer
Public NumDerivBC As Integer, NumDerivBCi As Integer, NumKwnPhi As Integer
Public NumKwnPhii As Integer, NumTimeSteps As Integer, NumTimeStepsi As Integer
Public SubKwnDBCi As Integer, MValKwnDBCi As Single, SValKwnDBCi As Single
Public Theta As Single, Thetai As Single, DeltaTime As Single, DeltaTimei As Single
Public MatlValues() As Single, Coord() As Single
Public InitialValue() As Single, EleNodeData() As Integer, EleMatl() As Integer
Public SubKwnSource() As Integer, SubKwnDBC() As Integer, SubKwnPhi() As Integer
Public ValKwnSource() As Single, MValKwnDBC() As Single,  SValKwnDBC() As Single
Public EFV() As Single, ESM() As Single, ECM() As Single, EleSub() As Integer
Public BandWidth As Integer, Diff As Integer
Public Const NumEleSub As Integer = 2, Const NumEleNodes As Integer = 2
Public NumNodalVal As Integer, NodalValuePerNode As Integer, ScrnWriteControl As Integer
Public FileWriteControl As Integer, NumTimeStepso As Integer, txtTime As Integer
Public Timet As Single, MigPercent As Single
Public Clb As ColorConstants, Clc As ColorConstants, Clr As ColorConstants, Clbl As ColorConstants,
Cly As ColorConstants, Clm As ColorConstants
Public txtCorelayer As Single, txtOutlayer As Single, txtExtralayer As Single, txtTotalMil As Single
Public Thnel As Integer, Thnod As Integer, Thinc As Single, TTh As Single, Thcl As Single, Thol As
Single, Thxl As Single, TThcoord As Single, txtTotalThick As Single,
Public IniArea As Single, InoArea As Single
Public WtPlastic As Single, VoFood As Single, txtSurArea As Single
Public DenFood As Single, WtFood As Single, PartitionC As Single, DenPlastic As Single
Public WtPlasticS As Single, VoFoodS As Single, VoPlastic As Single
Public WtFoodS As Single, IniConcPPM As Single, ConversionK As Single, ConcSubEqu As Single
Public MultiFactor As Single, MigPpm As Single, TimeSqrt As Single
Public ConcSubIni As Single, MaxMigConc As Single, ConcSubFood As Single, ConcSubVial As Single
Public TimeCalcSim As Integer, TimeSimOnly As Integer
Public NumMigData As Integer, MaxXax As Single, DeltaTimeHr As Single
Public AnyTimeT() As Single, AnyConc() As Single, GrpNumTimeS As Single, GrpTimet() As Single
Public GrpMigPer() As Single

Type itemstruc
   NumNodesi  As Integer : NumElei As Integer: NumMatlSetsi As Integer
   NumKwnPhii As Integer: SubKwnPhii(1 To 4) As Integer: txtCorelayer As Single
   txtOutlayer As Single: txtTotalThick As Single: txtExtralayer As Single
   txtTotalMil As Single: txtDxi(1 To 3) As Single: txtGi(1 To 3) As Single
   txtQi(1 To 3) As Single: txtDti(1 To 3) As Single: Coordi(1 To 15) As Single
   InitialValuei(1 To 15) As Single: EleNodeDatai(1 To 14) As Integer: EleNodeDataj(1 To 14) As Integer
   EleMatli(1 To 14) As Integer: WtPlastic As Single: DenPlastic(1 To 3) As Single
   txtSurArea As Single: VoFood As Single: DenFood As Single
   IniConcPPM As Single: PartitionC As Single: WtPlasticS As Single
   VoFoodS As Single: ConcSubFood As Single: ConcSubIni As Single
   ConcSubVial As Single: WtFood As Single: WtFoodS As Single
   Thetai As Single: DeltaTimei As Single: AnyTimeT(1 To 18) As Single: AnyConc(1 To 18) As Single
End Type
```

"BandMtx5" Code

```
'This module contains the subprograms related to the modification and solution of a system of equations
'that are symmetrical and banded. The coefficients in the stiffness matrix are stored in a rectangular array.

DefInt I-N

Sub BuildBandedSystem(NumEleSub%, EleSub%(), EFV(), ESM(), GFV(), GSM())
'incorporates the element force vector and the element stiffness matrix into the global force matrix and the
'global stiffness matrix.
'assumes that the global stiffness matrix is symmetrical and stored in a rectangular format. The direct
'stiffness method for a banded system of equations
For I = 1 To NumEleSub
 II = EleSub%(I)
 If (II > 0) Then
  GFV(II) = GFV(II) + EFV(I)
  For J = 1 To NumEleSub
   JJ = EleSub%(J)
   JJ = JJ - II + 1
   If (JJ > 0) Then
    GSM(II, JJ) = GSM(II, JJ) + ESM(I, J)
   End If
  Next J
 End If
Next I
End Sub

Sub CalBandWidth1D(NumEle%, EleNodeData%(), NodalValuePerNode%, BandWidth%)
MaxDiff = 0
  For I = 1 To NumEle%
    Diff% = Abs(EleNodeData%(I, 1) - EleNodeData%(I, 2))
    If (Diff% > MaxDiff) Then MaxDiff = Diff%
  Next I
BandWidth% = (MaxDiff + 1) * NodalValuePerNode%
End Sub

Sub DecompBandMatrix(NumNodalVal%, BandWidth%, GSM())
'decomposes a symmetric banded matrix into an upper triangular form using the method of Gaussian
'elemination. The matrix is stored in the rectangular array GSM(NumNodalVal%, BandWidth%). Only the
'upper part of the banded matrix is stored.
'Decompose the global stiffness matrix stored in a rectangular format
For I = 1 To (NumNodalVal - 1)
 MJ = I + BandWidth% - 1
 If (MJ > NumNodalVal) Then
  MJ = NumNodalVal
 End If
 NJ = I + 1
 MK = BandWidth%
 If ((NumNodalVal - I + 1) < BandWidth%) Then
  MK = NumNodalVal - I + 1
 End If
 ND = 0
 For J = NJ To MJ
  MK = MK - 1
  ND = ND + 1
  NL = ND + 1
```

```
    For K = 1 To MK
      NK = ND + K
      GSM(J, K) = GSM(J, K) - GSM(I, NL) * GSM(I, NK) / GSM(I, 1)
    Next K
  Next J
Next I
End Sub

Sub ModifyForceVector(NumKwnForce%, SubKwnForce%(), ValKwnForce(), GFV())
'modifies the global force vector, GFV(), by adding the known force/source values to it.
For I = 1 To NumKwnForce
  II = SubKwnForce%(I)
  GFV(II) = GFV(II) + ValKwnForce(I)
Next I
End Sub

Sub ModifyStiffMatrix(NumNodalVal, BandWidth%, SubKwnBdyCd%, ValKwnBdyCd, GSM(), GFV())
'modifies the global stiffness matrix by incorporating the known nodal. The method of deletion of rows and
'columns is used. Incorporation of the known boundary value
IB = SubKwnBdyCd%
K = IB - 1
For J = 2 To BandWidth%
  M = IB + J - 1
  If (M <= NumNodalVal) Then
    GFV(M) = GFV(M) - GSM(IB, J) * ValKwnBdyCd
    GSM(IB, J) = 0!
  End If
  If (K > 0) Then
    GFV(K) = GFV(K) - GSM(K, J) * ValKwnBdyCd
    GSM(K, J) = 0!
    K = K - 1
  End If
Next J
GFV(IB) = GSM(IB, 1) * ValKwnBdyCd
End Sub

Sub MultpyBandMatrix(NumNodalValues%, BandWidth%, GSM(), GFV(), ProdVector())
'multiplies a symmetric banded matrix and a column vector. The banded matrix is stored as a rectangular
'array and only the coefficients on and above the main diagonal are stored in the array.
For I = 1 To NumNodalValues
  Sum = 0!
  K = I - 1
  For J = 2 To BandWidth%
    M = J + I - 1
    If (M <= NumNodalValues) Then
      Sum = Sum + GSM(I, J) * GFV(M)
    End If
    If (K > 0) Then
      Sum = Sum + GSM(K, J) * GFV(K)
      K = K - 1
    End If
  Next J
  ProdVector(I) = Sum + GSM(I, 1) * GFV(I)
Next I
End Sub
```

```
Sub SolveBandMatrix(NumNodalVal%, BandWidth%, GSV(), GFV(), GSM())
'solves the system of banded equations using the method of Gaussian elimination.  The stiffness matrix has
'been decomposed prior to entering this subprogram using DecomposeBandMatrix.
'decomposes the column vector GFV(NumNodalVal%) before solving the system using backward
'substitution. Decompose the global force vector
For I = 1 To (NumNodalVal - 1)
  MJ = I + BandWidth% - 1
  If (MJ > NumNodalVal) Then
    MJ = NumNodalVal
  End If
  NJ = I + 1
  L = 1
  For J = NJ To MJ
    L = L + 1
    GFV(J) = GFV(J) - GSM(I, L) * GFV(I) / GSM(I, 1)
  Next J
Next I

'Solution by backward substitution
GSV(NumNodalVal) = GFV(NumNodalVal) / GSM(NumNodalVal, 1)
For K = 1 To (NumNodalVal - 1)
  I = NumNodalVal - K
  MJ = BandWidth%
  If ((I + BandWidth% - 1) > NumNodalVal) Then
    MJ = NumNodalVal - I + 1
  End If
  Sum = 0!
  For J = 2 To MJ
    N = I + J - 1
    Sum = Sum + GSM(I, J) * GSV(N)
  Next J
  GSV(I) = (GFV(I) - Sum) / GSM(I, 1)
Next K
End Sub

Sub ZeroGlobalMatrices(NumNodalVal%, BandWidth%, GFV(), GSM())
'fills the global stiffness matrix and force vector with zero values. This subprogram assumes that the global
'stiffness matrix is symmetrical and stored in a rectangular format.
For I = 1 To NumNodalVal
  GFV(I) = 0!
  For J = 1 To BandWidth%
    GSM(I, J) = 0!
  Next J
Next I
End Sub
```

"TimeMtx5" code

```
'This module contains the subprogram related to a finite element analysis in time.

Option Explicit
DefInt I-N

Sub BuildBandedTime(NumEleSub%, EleSub%(), EFV(), ECM!(), ESM!(), GFV(), GCM(), GSM())
'incorporates the element force vector, the element capacitance and stiffness matrices into the global
matrices for the time dependent problem.
'assumes that the global matrices are symmetrical and stored in a rectangular format.
For I = 1 To NumEleSub
 II = EleSub(I)
 If (II > 0) Then
  GFV(II) = GFV(II) + EFV(I)
  For J = 1 To NumEleSub
   JJ = EleSub(J)
   JJ = JJ - II + 1
   If (JJ > 0) Then
    GSM(II, JJ) = GSM(II, JJ) + ESM(I, J)
    GCM(II, JJ) = GCM(II, JJ) + ECM(I, J)
   End If
  Next J
 End If
Next I
End Sub

Sub ConvertCandK(NumNodes%, BandWidth%, Theta, DeltaTime, GCM(), GSM(), GFV())
'converts the [C] and [K] matrices to [A] and [P] using [A] = [C] + Theta*DeltaTime*[K]
'[P] = [C] - (1-Theta)*DeltaTime*[K]
Dim AA As Single, BB As Single, TC As Single, TK As Single
AA = Theta * DeltaTime
BB = (1 - Theta) * DeltaTime
For I = 1 To NumNodes%
 For J = 1 To BandWidth%
  TC = GCM(I, J)
  TK = GSM(I, J)
  GCM(I, J) = TC + AA * TK
  GSM(I, J) = TC - BB * TK
 Next J
Next I
End Sub

Sub ConvertFvector(NumNodes%, Theta, DeltaTime, GFV())
'converts the global force vector GFV() using GFV() = DeltaTime*GFV()
For I = 1 To NumNodes%
 GFV(I) = GFV(I) * DeltaTime
Next I
End Sub

Sub ModifyCandK(NumNodalVal%, BandWidth%, NumKwnPhi%, SubKwnPhi%(), GlobalPhiOld(),
GFV(), GSM(), GCM())
'modifies [C] and [K] such that the nodal values that are not a function of time remain constant on each
'iteration.
Dim N As Integer, JM As Integer, M As Integer
For I = 1 To NumKwnPhi%
```

```
 J = SubKwnPhi%(I)
 N = J - 1
 For JM = 2 To BandWidth%
  M = J + JM - 1
  If (M <= NumNodalVal%) Then
   GFV(M) = GFV(M) - GSM(J, JM) * GlobalPhiOld(J)
   GSM(J, JM) = 0
   GCM(J, JM) = 0
  End If
  If (N > 0) Then
   GFV(N) = GFV(N) - GSM(N, JM) * GlobalPhiOld(J)
   GSM(N, JM) = 0
   GCM(N, JM) = 0
   N = N - 1
  End If
 Next JM
 GCM(J, 1) = 1
 GSM(J, 1) = 0
 GFV(J) = 0
Next I
End Sub

Sub ZeroGlobalMtxTime(NumNodalVal%, BandWidth%, GFV(), GCM(), GSM())
'fills the global capacitance matrix, the global stiffness matrix and force vector with zero values.
'assumes that both global matrices are symmetrical and are stored in a rectangular format.
For I = 1 To NumNodalVal
 GFV(I) = 0!
 For J = 1 To BandWidth%
  GCM(I, J) = 0!
  GSM(I, J) = 0!
 Next J
Next I
End Sub
```

"Output5" Code

```
Sub DefineGlobalPhi(NumNodalVal%, Value!(), GlobalPhi())
'This subprogram places the values in Value() into GlobalPhi()
For I = 1 To NumNodalVal
 GlobalPhi(I) = Value(I)
Next I
End Sub
```

APPENDIX C

DATA

Table 6.1 BHT migration to 100% ethanol from single core layer structure (effective thickness: 0.6 mil)

40°C			31°C			23°C		
t, hrs	$t^{0.5}$	ppm*, g/cm^3	t, hrs	$t^{0.5}$	ppm*, g/cm^3	t, hrs	$t^{0.5}$	ppm*, g/cm^3
0.0	0.00	0.00	0.0	0.00	0.00	0.0	0.00	0.00
0.5	0.71	25.98	1.0	1.00	23.80	1.0	1.00	11.15
1.0	1.00	36.85	2.0	1.41	34.21	3.0	1.73	21.00
2.0	1.41	48.52	3.0	1.73	40.05	6.5	2.55	31.60
3.0	1.73	53.32	6.5	2.55	52.14	14.0	3.74	45.06
4.0	2.00	55.80	14.0	3.74	56.00	18.5	4.30	49.50
6.5	2.55	55.83	18.5	4.30	55.30	36.0	6.00	54.43
14.0	3.74	56.30	36.0	6.00	56.20	71.5	8.46	55.50
27.0	5.20	55.92	71.5	8.46	57.00	117.5	10.84	55.68
36.0	6.00	55.56	117.5	10.84	56.44	147.0	12.12	55.85
65.5	8.09	56.12	147.0	12.12	55.80	210.0	14.49	55.76
							16.20	55.70

* Average of 3 or more measurements.

Table 6.2 BHT migration to 100% ethanol from multilayer structure with an effective thickness ratio of 1:1 (0.6/0.6 mil)

40°C			31°C			23°C		
t, hrs	$t^{0.5}$	ppm*, g/cm^3	t, hrs	$t^{0.5}$	ppm*, g/cm^3	t, hrs	$t^{0.5}$	ppm*, g/cm^3
0.0	0.00	0.00	0.0	0.00	0.00	0.0	0.00	0.00
2.0	1.41	20.93	3.0	1.73	15.41	3.0	1.73	7.89
3.5	1.87	31.90	6.5	2.55	31.84	14.0	3.74	22.13
9.0	3.00	62.54	17.5	4.18	52.35	29.0	5.39	34.23
21.0	4.58	74.45	41.0	6.40	70.60	69.0	8.31	55.39
27.0	5.20	77.13	74.5	8.63	76.18	115.0	10.72	64.45
65.5	8.09	78.78	117.5	10.84	78.60	162.0	12.73	71.33
112.5	10.61	77.48	147.0	12.12	78.55	210.0	14.49	75.29
143.5	11.98	77.74	213.5	14.61	78.62	262.5	16.20	77.15
195.0	13.96	78.43	334.0	18.28	78.48	337.0	18.36	79.93
262.0	16.19	78.16	404.0	20.10	78.27	433.5	20.82	79.79
335.0	18.30	78.68	504.5	22.46	78.82	548.5	23.42	78.89
						668.0	25.85	78.42
						770.0	27.75	78.42
						858.5	29.30	78.82

* Average of 3 or more measurements.

Table 6.3 BHT migration to 100% ethanol from multilayer structure with an effective thickness ratio of 1:2 (0.6/1.2 mil)

40°C			31°C			23°C		
t, hrs	$t^{0.5}$	ppm*, g/cm^3	t, hrs	$t^{0.5}$	ppm*, g/cm^3	t, hrs	$t^{0.5}$	ppm*, g/cm^3
0.0	0.00	0.00	0.0	0.00	0.00	0.0	0.00	0.00
2.0	1.41	5.67	3.0	1.73	2.90	6.5	2.55	1.97
6.5	2.55	27.0	6.5	2.55	10.20	14.0	3.74	6.45
9.0	3.00	40.67	17.5	4.18	26.88	29.0	5.39	15.03
21.0	4.58	68.45	41.0	6.40	52.49	69.0	8.31	33.52
52.4	7.24	82.61	74.5	8.63	69.27	115.0	10.72	45.80
112.5	10.61	86.18	142.5	11.94	83.04	162.0	12.73	56.17
172.5	13.13	87.11	213.5	14.61	86.50	210.0	14.49	65.33
264.5	16.26	87.35	334.0	18.28	87.38	262.5	16.20	71.11
381.5	19.53	87.18	404.0	20.10	86.80	337.0	18.36	77.66
			504.5	22.46	86.75	433.5	20.82	83.61
						548.5	23.42	87.93
						668.0	25.85	87.41
						770.0	27.75	87.41
						858.5	29.30	87.67

* Average of 3 or more measurements.

Table 6.4 BHT migration to 100% ethanol from multilayer structure with an effective thickness ratio of 1:3 (0.6/1.8 mil)

40°C			31°C			23°C		
t, hrs	$t^{0.5}$	ppm*, g/cm^3	t, hrs	$t^{0.5}$	ppm*, g/cm^3	t, hrs	$t^{0.5}$	ppm*, g/cm^3
0.0	0.00	0.00	0.0	0.00	0.00	0.0	0.00	0.00
2.0	1.41	1.21	3.0	1.73	0.30	14.0	3.74	0.00
9.0	3.00	19.80	6.5	2.55	2.05	29.0	5.39	4.69
21.0	4.57	45.05	17.5	4.18	9.50	69.0	8.31	15.69
52.5	7.25	69.12	41.0	6.40	27.28	115.0	10.72	24.17
112.5	10.61	79.89	74.5	8.63	45.85	162.0	12.73	33.49
195.0	13.96	84.37	142.5	11.94	67.27	210.0	14.49	40.97
262.0	16.19	84.44	213.5	14.61	77.02	262.5	16.20	47.71
335.0	18.30	84.54	334.0	18.28	83.93	337.0	18.36	56.02
433.5	20.82	84.01	404.0	20.10	84.11	433.5	20.82	63.72
			504.5	22.46	83.31	548.5	23.42	69.41
			668.0	25.85	84.05	668.0	25.85	73.55
						858.0	29.30	77.58
						1124.0	33.53	81.85
						1291.0	35.93	83.05
						1505.0	38.79	83.77
						1725.0	41.53	83.55

* Average of 3 or more measurements.

Table 6.5 BHT migration to 50% ethanol from single core layer structure (effective thickness: 0.6 mil)

40°C			31°C			23°C		
t, hrs	$t^{0.5}$	ppm*, g/cm^3	t, hrs	$t^{0.5}$	ppm*, g/cm^3	t, hrs	$t^{0.5}$	ppm*, g/cm^3
0.0	0.00	0.00	0.0	0.00	0.00	0.0	0.00	0.00
0.5	0.71	17.80	1.0	1.00	16.22	1.0	1.00	10.78
1.0	1.00	26.50	2.0	1.41	23.00	3.0	1.73	18.33
2.0	1.41	35.01	3.0	1.73	28.59	6.0	2.45	26.03
3.0	1.73	40.60	6.0	2.45	39.16	10.5	3.24	32.17
6.0	2.45	49.86	9.0	3.00	45.30	26.5	5.15	41.24
9.0	3.00	50.94	20.5	4.53	48.42	52.0	7.21	43.89
20.8	4.56	49.80	48.0	6.93	48.60	107.5	10.37	47.16
27.0	5.20	51.04	74.5	8.63	48.47	143.5	11.98	47.38
52.3	7.23	50.93	142.5	11.94	48.75	196.5	14.02	47.41
72.0	8.49	50.70	217.5	14.75	48.00	263.5	16.23	47.38

* Average of 3 or more measurements

Table 6.6 BHT migration to 50% ethanol from multilayer structure with an effective thickness ratio of 1:1 (0.6/0.6 mil)

40°C			31°C			23°C		
t, hrs	$t^{0.5}$	ppm*, g/cm^3	t, hrs	$t^{0.5}$	ppm*, g/cm^3	t, hrs	$t^{0.5}$	ppm*, g/cm^3
0.0	0.00	0.00	0.0	0.00	0.00	0.0	0.00	0.00
2.0	1.41	14.89	3.0	1.73	11.61	10.5	3.24	12.64
4.0	2.00	28.56	6.5	2.55	21.37	26.5	5.15	21.42
9.0	3.00	44.07	13.5	3.67	30.43	52.0	7.21	31.19
21.0	4.58	59.64	31.0	5.57	48.45	107.5	10.37	43.38
52.5	7.25	64.00	55.5	7.45	55.89	143.5	11.98	49.53
85.0	9.22	65.21	82.5	9.08	58.25	196.5	14.02	52.88
188.5	13.73	64.87	131.5	11.47	60.94	263.5	16.23	55.55
266.5	16.32	64.57	196.5	14.02	61.55	343.5	18.53	57.16
382.0	19.54	65.38	266.5	16.32	61.95	436.5	20.89	57.74
			343.5	18.53	61.60	511.5	22.62	57.34
			436.0	20.88	61.10	701.5	26.49	57.97
						803.5	28.35	57.18

* Average of 3 or more measurements

Table 6.7 BHT migration to 50% ethanol from multilayer structure with an effective thickness ratio of 1:2 (0.6/1.2 mil)

40°C			31°C			23°C		
t, hrs	$t^{0.5}$	ppm*, g/cm^3	t, hrs	$t^{0.5}$	ppm*, g/cm^3	t, hrs	$t^{0.5}$	ppm*, g/cm^3
0.0	0.00	0.00	0.0	0.00	0.00	0.0	0.00	0.00
2.0	1.41	3.69	3.0	1.73	2.97	10.5	3.24	3.84
4.0	2.00	10.88	6.0	2.45	6.88	26.5	5.15	7.64
9.0	3.00	25.02	20.5	4.53	18.97	52.0	7.21	12.95
22.0	4.69	46.46	48.0	6.93	36.86	107.5	10.37	24.41
52.5	7.25	63.33	74.5	8.63	46.75	143.5	11.98	30.55
82.5	9.08	66.78	142.5	11.94	57.01	196.5	14.02	37.05
188.5	13.73	67.89	217.5	14.75	61.64	263.5	16.23	42.45
266.5	16.32	68.16	334.5	18.29	62.80	343.5	18.53	47.56
382.5	19.56	68.47	407.5	20.19	61.81	436.5	20.89	50.07
			521.5	22.84	62.61	511.5	22.62	51.84
			598.5	24.46	62.38	701.5	26.49	56.12
						803.5	28.35	55.26
						1008.5	31.76	57.16
						1134.5	33.68	57.67
						1344.5	36.67	56.84

* Average of 3 or more measurements

Table 6.8 BHT migration to 50% ethanol from multilayer structure with an effective thickness ratio of 1:3 (0.6/1.8 mil)

40°C			31°C			23°C		
t, hrs	$t^{0.5}$	ppm*, g/cm^3	t, hrs	$t^{0.5}$	ppm*, g/cm^3	t, hrs	$t^{0.5}$	ppm*, g/cm^3
0.0	0.00	0.00	0.0	0.00	0.00	0.0	0.00	0.00
2.0	1.41	1.52	6.5	2.55	2.13	10.5	3.24	1.46
9.0	3.00	11.96	13.5	3.67	4.67	26.5	5.15	2.31
13.0	3.61	17.45	31.0	5.57	13.98	52.0	7.21	5.08
20.8	4.56	25.72	55.5	7.45	24.00	107.5	10.37	9.82
40.5	6.36	44.56	82.5	9.08	33.32	143.5	11.98	13.25
60.0	7.75	52.61	131.5	11.47	44.92	196.5	14.02	18.37
82.5	9.08	58.55	196.5	14.02	52.02	263.5	16.23	23.72
131.5	11.47	61.88	266.5	16.32	54.31	343.5	18.53	28.47
155.5	12.47	61.25	343.5	18.53	53.86	436.5	20.89	32.39
231.5	15.22	61.50	436.0	20.88	52.55	511.5	22.62	36.17
334.2	18.30	61.63	521.5	22.84	52.81	701.5	26.49	41.84
						803.5	28.35	43.98
						961.5	31.01	46.54
						1134.5	33.68	47.96
						1344.5	36.67	48.73
						1564.5	39.55	49.23

* Average of 3 or more measurements

Table 6.9 Irganox™1076 migration to 100% ethanol from single core layer structure (effective thickness: 0.6 mil)

40°C			31°C			23°C		
t, hrs	$t^{0.5}$	ppm*, g/cm^3	t, hrs	$t^{0.5}$	ppm*, g/cm^3	t, hrs	$t^{0.5}$	ppm*, g/cm^3
0.0	0.00	0.00	0.0	0.00	0.00	0.0	0.00	0.00
1.0	1.00	15.41	3.0	1.73	12.10	21.0	4.58	15.27
3.0	1.73	27.79	7.0	2.65	19.74	50.0	7.07	24.41
7.0	2.65	42.18	21.0	4.58	34.19	93.0	9.64	33.00
21.0	4.58	63.08	50.0	7.07	47.33	165.5	12.86	41.13
50.0	7.07	64.65	93.0	9.64	58.38	236.0	15.36	48.90
93.0	9.64	65.69	165.5	12.86	63.49	308.5	17.56	53.84
150.0	12.25	65.12	236.0	15.36	65.20	408.5	20.21	59.17
236.0	15.36	65.60	308.5	17.56	64.84	501.5	22.39	61.08
308.5	17.56	66.12	408.5	20.21	65.61	622.0	24.94	63.20
408.5	20.21	65.91	501.5	22.39	65.78	794.5	28.19	65.89
501.5	22.39	65.59	622.0	24.94	64.95	1057.5	32.52	65.59
622.0	24.94	64.70	794.5	28.19	65.28	1437.0	37.91	64.98
794.5	28.19	65.94	1057.5	32.52	65.98	1657.5	40.71	64.92
1057.5	32.52	66.54	1437.0	37.91	65.99	1967.5	44.36	65.75
1437.0	37.91	66.21	1657.5	40.71	66.11	2256.5	47.50	65.33
1657.5	40.71	66.35	1967.5	44.36	66.05	2664.0	51.61	65.88
1967.5	44.36	66.39	2256.5	47.50	64.82			
2256.5	47.50	66.12	2664.0	51.61	65.12			
2664.0	51.61	65.85						

* Average of 3 or more measurements

Table 6.10 Irganox™1076 migration to 100% ethanol from multilayer structure with an effective thickness ratio of 1:1 (0.6/0.6 mil)

40°C			31°C			23°C		
t, hrs	$t^{0.5}$	ppm*; g/cm^3	t, hrs	$t^{0.5}$	ppm*; g/cm^3	t, hrs	$t^{0.5}$	ppm*; g/cm^3
0.0	0.00	0.00	0.0	0.00	0.00	0.0	0.00	0.00
3.0	1.73	0.00	3.0	1.73	0.00	21.0	4.58	0.00
7.0	2.65	3.46	7.0	2.65	0.00	50.0	7.07	0.00
21.0	4.58	19.33	21.0	4.58	1.30	93.0	9.64	1.27
50.0	7.07	41.94	50.0	7.07	5.77	165.5	12.86	3.01
93.0	9.64	57.25	93.0	9.64	14.45	236.0	15.36	4.84
150.0	12.25	65.83	165.5	12.86	23.39	308.5	17.56	7.13
236.0	15.36	72.15	236.0	15.36	33.81	408.5	20.21	10.35
308.5	17.56	75.63	308.5	17.56	40.31	501.5	22.39	12.98
408.5	20.21	75.78	408.5	20.21	46.27	622.0	24.94	16.16
501.5	22.39	75.61	501.5	22.39	49.68	794.5	28.19	21.41
622.0	24.94	75.15	622.0	24.94	53.95	1057.5	32.52	26.45
794.5	28.19	74.69	794.5	28.19	59.38	1437.0	37.91	37.09
1057.5	32.52	74.56	1057.5	32.52	64.51	1657.5	40.71	41.03
1437.0	37.91	75.36	1437.0	37.91	71.38	1967.5	44.36	45.51
1657.5	40.71	74.83	1657.5	40.71	73.56	2256.5	47.50	48.25
1967.5	44.36	76.05	1967.5	44.36	73.37	2664.0	51.61	51.87
2256.5	47.50	75.86	2256.5	47.50	75.58			
2664.0	51.61	75.71	2664.0	51.61	75.66			

* Average of 3 or more measurements

Table 6.11 Irganox™1076 migration to 100% ethanol from multilayer structure with an effective thickness ratio of 1:2 (0.6/1.2 mil)

40°C			31°C			23°C		
t, hrs	$t^{0.5}$	ppm*, g/cm^3	t, hrs	$t^{0.5}$	ppm*, g/cm^3	t, hrs	$t^{0.5}$	ppm*, g/cm^3
0.0	0.00	0.00	0.0	0.00	0.00	0.0	0.00	0.00
3.0	1.73	0.00	3.0	1.73	0.00	21.0	4.58	0.00
7.0	2.65	0.00	7.0	2.65	0.00	50.0	7.07	0.00
21.0	4.58	1.91	21.0	4.58	0.00	93.0	9.64	0.00
50.0	7.07	12.43	50.0	7.07	0.00	165.5	12.86	0.00
93.0	9.64	26.05	93.0	9.64	1.16	236.0	15.36	0.00
150.0	12.25	41.93	165.5	12.86	4.76	308.5	17.56	0.00
236.0	15.36	54.61	236.0	15.36	8.49	408.5	20.21	0.00
308.5	17.56	60.85	308.5	17.56	12.51	501.5	22.39	0.00
408.5	20.21	64.85	408.5	20.21	17.16	622.0	24.94	1.07
501.5	22.39	67.50	501.5	22.39	20.66	794.5	28.19	2.91
622.0	24.94	68.55	622.0	24.94	25.40	1057.5	32.52	4.79
794.5	28.19	69.66	794.5	28.19	31.01	1437.0	37.91	10.06
1057.5	32.52	71.51	1057.5	32.52	40.33	1657.5	40.71	12.48
1437.0	37.91	71.35	1437.0	37.91	51.34	1967.5	44.36	16.39
1657.5	40.71	71.89	1657.5	40.71	54.68	2256.5	47.50	19.19
1967.5	44.36	71.95	1967.5	44.36	60.64	2664.0	51.61	22.54
2256.5	47.50	72.97	2256.5	47.50	63.24			
2664.0	51.61	73.15	2664.0	51.61	66.61			

*Average of 3 or more measurements

Table 6.12 Irganox™1076 migration to 100% ethanol from multilayer structure with an effective thickness ratio of 1:3 (0.6/1.8 mil)

40°C			31°C			23°C		
t, hrs	$t^{0.5}$	ppm*, g/cm^3	t, hrs	$t^{0.5}$	ppm*, g/cm^3	t, hrs	$t^{0.5}$	ppm*, g/cm^3
0.0	0.00	0.00	0.0	0.00	0.00	0.0	0.00	0.00
3.0	1.73	0.00	3.0	1.73	0.00	21.0	4.58	0.00
7.0	2.65	0.00	7.0	2.65	0.00	50.0	7.07	0.00
21.0	4.58	0.00	21.0	4.58	0.00	93.0	9.64	0.00
50.0	7.07	2.67	50.0	7.07	0.00	165.5	12.86	0.00
93.0	9.64	9.45	93.0	9.64	0.00	236.0	15.36	0.00
150.0	12.25	18.10	165.5	12.86	0.00	308.5	17.56	0.00
236.0	15.36	29.14	236.0	15.36	0.00	408.5	20.21	0.00
308.5	17.56	36.00	308.5	17.56	1.90	501.5	22.39	0.00
408.5	20.21	42.81	408.5	20.21	4.68	622.0	24.94	0.00
501.5	22.39	47.70	501.5	22.39	7.27	794.5	28.19	0.00
622.0	24.94	51.61	622.0	24.94	10.88	1057.5	32.52	0.00
794.5	28.19	56.40	794.5	28.19	14.07	1437.0	37.91	1.35
1057.5	32.52	59.59	1057.5	32.52	19.88	1657.5	40.71	2.60
1437.0	37.91	62.94	1437.0	37.91	29.05	1967.5	44.36	3.65
1657.5	40.71	63.97	1657.5	40.71	33.30	2256.5	47.50	4.97
1967.5	44.36	65.51	1967.5	44.36	39.26	2664.0	51.61	7.80
2256.5	47.50	66.38	2256.5	47.50	41.76			
2664.0	51.61	66.49	2664.0	51.61	47.25			

* Average of 3 or more measurements

BIBLIOGRAPHY

BIBLIOGRAPHY

Adamson, A. W. and Gast, A. P. (1997). Chapter XI. The solid-liquid interface - adsorption from solution, Physical Chemistry of Surfaces, Sixth Edition, John Wiley & Sons, Inc. 390-430.

American Plastics Council, (1999, September). Plastics Recycling Markets Database, R.W. Beck, Inc.

Arthur D. Little, Inc. (1983). A study of indirect food additive migration, Final Summary Report. FDA contract number 223-77-2360, Washington, DC.

ASTM D 4754-93, (1995). Annual Book of ASTM Standards, Section 8, Volume 08.03, ASTM, PA, USA, 178-181.

Bakker, M. (1994, May). Using recycled plastics in food bottles: the technical barriers, Resource Recycling, 59-64.

Baner, A. L., Franz, R. and Piringer, O. (1994). Alternative methods for the determination and evaluation of migration potential from polymeric food contact materials, Deutsche Lehensmittel-Rundschau, 90, 137-143, and 90, 181-185.

Baner, A., Bransch, J., Franz, R. and Piringer, O. (1996). The application of a predictive migration model for evaluating the compliance of plastic materials with European food regulations, Food Additives and Contaminants, 13, 587-601.

Barlas, S. (1995, June). FDA plans to speed approval for PCR use, Packworld, 24-27.

Bayer, F.L. (1997). The threshold of regulation and its application to indirect food additive contaminants in recycled plastics, Food Additives and Contaminants,14, 6-7: 661-670.

Becker, K., Koszinowski, J. and Piringer, O. (1983). Permeation of flavor and aromas through polyolefins, Deutsche Lehensmittel-Rundschau, 79(8), 257-266.

Begley, T.H. and Hollifield, H.C. (1993, November). Recycled polymers in food packaging: migration considerations, Food Technology, 109-112.

Begley, T.M. (1997). Methods and approaches used by FDA to evaluate the safety of food packaging materials, Food Additives and Contaminants, Vol. 4., No 6-7, 545-553.

Boven, G., Brinkhuis, R.H.G., Vorenkamp, E.J. and Schouten, A.J. (1991). Interdiffusion of thin polymer layers studied by external reflection infrared spectroscopy, Macromolecules, 24, 967-969.

Carslaw, H.S. and Jaeger, J.C. (1959). Conduction of Heat in Solids, 2nd Edition, Clarendon Press, Oxford, England, 319-326

Castle, L., Franz, R., Huber, M., Piringer, O., Damant, A. and Jickells, S. (1996). Study of functional barrier properties of multilayer recycled poly (ethylene terephthalate) bottles for soft drinks, Journal of Agricultural and Food Chemistry, 44, 892-897.

Chatwin, P.C. and Katan, L.L. (1989). The role of mathematics and physics in migration predictions, Packaging Technology and Science, 2, 75-84.

Crank, J. (1975). The Mathematics of Diffusion, 2nd Edition, Clarendon Press, Oxford, England, 44-60.

Doyle, M. (1996). Packaging, solid waste, and environmental trade-offs, Packaging Strategy, Technomic Publishing Co. Lancaster, PA, 23-34

Environmental Protection Agency, (2001). Municipal Solid Waste in the United States: 1999 Facts and Figures, EPA 530-R-01-014, Washington DC. 1-17

Etters, J.N. (1991). Mass transport in finite baths: effect of surface barriers. Industrial & Engineering Chemistry Research, 30, 589-591.

Figge, K. (1972). Migration of additives from plastics films into edible oils and fat simulants, Food and Cosmetics Toxicology, 10, 815-828.

Figge, K. (1980). Migration of components from plastics packaging materials into packed goods - test methods and diffusion models. Progress Polymer Science, 6, 187-252.

Figge, K. (1996). Plastic packages for foodstuffs: A topical survey of legal regulations and migration testing, NATEC Institut fur naturwissenschaftlich-technische Diente GmbH, Hamburg, Germany, 10-19.

Food and Drug Administration, (1992). Points to consider for the use of recycled plastics in food packaging: chemistry considerations. Food and Drug Administration, Indirect Aadditive Branch, HFS-216, Washington, DC.

Food and Drug Administration, (1993). Food additives: threshold of regulation for substances used in food-contact articles: 21 CFR. Federal Register, 58, 52719-52729.

Food and Drug Administration, (1995, July 17). Federal Register, 60, 36582-36596.

Food and Drug Administration, (1995). Recommendations for chemistry data for indirect food additive petitions, Chemistry Review Branch, Office of Premarket Approval, Center for Food Safety & Applied Nutrition, Washington, D.C.

Ford, T. (1994, June 20). Continental making hot-fill, post-consumer PET bottle, Plastics News, 1-20.

Franz, R., Huber, M. and Piringer, O.G. (1994). Testing and evaluation of recycled plastics for food packaging use – possible migration through a functional barrier, Food Additives and Contaminants, Vol. 11, No 4, 479-496.

Franz, R., Huber, M. and Piringer, O.G. (1997). Presentation and experimental verification of a physico-mathematical model describing the migration across functional barrier layers into foodstuffs, Food Additives and Contaminants, Vol. 14, No 6-7, 627-640.

Gandek, T.P., Hatton, T.A. and Reid, R.C. (1989). Batch extraction with reaction: phenolic antioxidant migration from polyolefin to water. I. Theory and batch extraction with reaction: phenolic antioxidant migration from polyolefins to water. 2. Experimental results and discussion. Industrial & Engineering Chemistry Research, 28, 1030- 1045.

Graff, G. (1992, July). Recycling HDPE bottles: good will is not enough. Modern Plastics International, 31-33.

Gavara, R., Hernandez, R.J. and Giacin, J. (1996). Methods to determine partition coefficient of organic compounds in water/polystyrene systems, Journal of Food Science, 61(5), 947-952.

Hamdani, M., Feigenbaum, A. and Vergnaud, J.M. (1997). Prediction of worst case migration from packaging to food using mathematical models, Food Additives and Contaminants, Vol.14, No 5, 499-506.

Ilter, M., Ozilgen, M. and Orbey, N. (1991). Modelling permeation of modified atmosphere gas mixtures through LD polyethylene package film, Polymer International, 25, 211-217.

Jabarlin, S.A. and Kollen, W.J. (1988). Polyolefin properties for rigid food packaging, Polymer Engineering and Science, 28, 1156-1161.

Katan, L.L. (1996). Do 'functional barriers' function?, Packaging Technology and Science. Vol 9, 289-296.

Khinava, R.S. and Aminabhavi, T.M. (1992). Resistance of barrier elastomers to hazardous organic liquids. Journal of Applied Polymer Science, 45, 1107-1125.

Kinigakis, P., Miltz, J. and Gilbert, S.G. (1987). Partition of VCM in PVC food simulant systems, Journal of Food Processing and Preservation, 11, 247-255.

Lai, Ch.C. and Selke, S.E. (1988). Developments and applications of recycled plastics (a conference report). Packaging Technology and Science, 1, 157-160.

Laoubi, S., Feigenbaum, A. and Vergnaud, J.M. (1995). Effect of functional barrier thickness for a food package with recycled polymer, Packaging Technology and Science, Vol 8, 249-259.

Laoubi, S. and Vergnaud, J.M. (1996). Theoretical treatment of pollutant transfer in a finite volume of food from a polymer packaging made of a recycled film and a functional barrier, Food Additives and Contaminants, Vol. 13, No 3, 293-306.

Laoubi, S. and Vergnaud, J.M. (1997). Modelling transport between a monolayer and a bi-layer polymeric package and food, Food Additives and Contaminants, Vol. 14, No 6-7, 641-647.

Lewis, R.W., Morgan, K. and Schrefler, B.A. (1981) Numerical methods in heat transfer, J. Wiley, Chichester [Eng]; New York.

Lewis, R.W., Morgan, K. Thomas, H.R. and Seetharamu, K.N. (1996). The finite element method in heat transfer analysis, J. Wiley, Chichester, [Eng]; New York.

Lickly, T.D., Bell, C.D. and Lehr, K.M. (1990). The Migration of Irganox 1010 antioxident from high-density polyethylene and polypropylene into a series of potential fatty-food simulants, Food Additives and Contaminants, 7, 6, 805-814.

Lingle, R. (1999, July). Miller fine-tunes plastic bottles, components, Packaging Digest, 42.

Linssen, J.P.H., Reistma, J.C.E. and Cozijnsen, J.L. (1998). Research note – migration of antioxidants from polyolefins into ethanolic simulants, Packaging Technology and Science, 11, 241-245.

Miller, C. (1993, June 29). Recycled PET approved by FDA for use in more food packages, Recycling Times.

Miltz, J. (1992). Food packaging. Handbook of Food Engineering, edited by D. Heldman and D. Lund, Marcel Dekker, Inc. New York, 667-718.

Miltz, J., Passy, N. and Mannhein, C.H. (1992). Mass transfer from and through packaging materials. Packaging Technology and Science, 5, 49-56.

National Bureau of Standards, (1982). Migration of low molecular weight additives in polyolefins and copolymers, Final Project Report, NBSIR 82-2472,", U.S. Department of Commerce.

Packworld, (1995, March). Coke's first contour bottle with PCR, 2.

Piringer, O.G. (1994). Evaulation of plastics for food packaging, Food Additives and Contaminants,11, 2, 221-230.

Piringer, O., Franz, R., Huber, M., Begley, T.H. and McNeal, T.P. (1998). Migration from food packaging containing a functional barriers: mathematical and experimental evaluation", J. Agric. Food Chem. 46, 1532-1538.

Plastics Recycling Task Force, (1994). Guide for the safe use of recycled plastics for food packaging application. National Food Processors Association & The Society of the Plastics Industry, Inc.Washington, DC.

Powell, J. (1995, May). HDPE bottle recycling: the demand glut continues, Resource Recycling, 23-28.

Powell, J. (1999, October). Beer in plastic: Recycling's minor bump or major barrier?, Resource Recycling, 23-26.

Reddy J.N. and Gartling D.K. (1994). The finite element method in heat transfer and fluid dynamics, CRC Press, Boca Raton.

Reid, R.C., Sidman, K.R., Schwope, A.D. and Till, D.E. (1980). Loss of adjuvants from polymer films to foods or food simulants: effects of the external phase, Ind. Eng. Chem. Prod. Res. Dev. 19: 580-587.

Reynier, A., Dole, P. and Feigenbaum, A. (1999) Prediction of worst case migration: presentation of a rigorous methodology, Food Additives and Contaminants,16, 4, 137-152.

Rudolph, F.B. (1979). Diffusion in a multicomponent inhomogeneous system with moving boundaries: 1. Swelling at constant volume, Journal of Polymer Science: Polymer Physics Edition, 17, 1709-1718.

Schwope, A.D., Ehntholt, D.J., Sidman, K.R., Whelan, R.H., Schwartz, P.S. and Reid, R.C. (1987). Migration of BHT and Irganox 1010 from LDPE to foods and food-simulating liquids, Food and Cosmetics Toxicology, 25, 4: 317-326.

Segerlind, L.J. (1984). Applied Finite Element Analysis, 2nd edition, John Wiley & Sons, Inc., New York.

Seyler, R.J. (1990). Slow rate penetration of packaging films. Journal of Plastic Film and Sheeting, 6, 191-224.

Thayer, A.M. (1989, January 30). Solid waste concerns spur plastic recycling efforts. Chemical and Engineering News, 7-15.

Thompson Publishing Group, (2001). Environmental Packaging, Washington, D.C.

Thorsheim, H. and Amstrong, D. (1993, August). Recycled plastics for food packaging, Chemtech, 55-58.

Vergnaud, J.M. (1991). Liquid Transport Processes in Polymeric Materials, Prentice Hall, Englewood Cliffs, NJ, 45-61.

Vergnaud, J.M. (1995/96). General survey on the mass transfer taking place between a polymer and a liquid. Polymer Engineering, 15, 57-77.