UCTAS - THE UCT ANAESTHETICS SIMULATOR
SIMULATING THE UPTAKE AND DISTRIBUTION OF HALOTHANE.

By

Robin Andrew Cooper

(Bsc Elec. Eng.)

Submitted to the University of Cape Town

in partial fulfilment of the requirements for the degree of

Master of Science in Medicine

in the field of Biomedical Engineering.

DECEMBER 1988

graph or line-graph form and may also be dumped to a text file for use by other plotting programs.

ACKNOWLEDGEMENTS

I would like to thank the following people who have made this work possible :

My first supervisor John Amoore who pointed me in the right direction and was ever helpfull in correcting my spelling.

My supervisor, Dr Dave Morrel, from the UCT medical school anaesthetics department for his advice and discussions on anaesthesia.

My other supervisor, Dr Tony Bunn, for his help and constructive critisism and for always having time to discuss any problems.

My friend Annalu Waller for her pascal lessons, programming ideas and encouragement.

I would like to thank the Medical Research Council for financing my first two years of study. I would also like to thank the staff and members of the Department of Biomedical Engineering at UCT for allowing me to take a 2 year holiday in the Antarctic in the middle of my degree.

Finally I would like to thank my parents for their unfailing support and understanding during the course of this degree.

ABSTRACT

An anaesthetic simulator program that runs on an IBM personal computer system has been developed. The program allows an operator to observe the uptake and distribution of the volatile anaesthetic agent halothane by a standard 75kg patient. The "patient's" breathing is assisted by a ventilator and the anaesthetic gas is supplied through a simulated circle breathing circuit.

The most important component of a simulator is a mathematical model of the system being simulated. In this case a model of the uptake and distribution of the anaesthetic agent halothane by the human cardiovascular and respiratory systems was required. Such a model was developed by combining features of several existing non-linear multi-compartmental models and adapting the equations to allow them to be implemented on a digital computer.

The simulator software that was developed allows an operator to adjust physical parameters such as fresh gas flow rate, halothane concentration, and breathing parameters from the keyboard of an IBM PC computer, and observe the way various model parameters respond on a graphics screen. The speed of the simulation is adjustable. ie, the state of the model can be repetitively calculated and displayed at 1, 10, or 60 second intervals. Model parameters can be displayed in bar-graph or line-graph form and may also be dumped to a text file for use by other plotting programs.

The software package developed should provide a useful teaching aid to anaesthetists, and to anyone desiring to understand the pharmacokinetics of the uptake and distribution of a volatile anaesthetic agent by a human patient.

TABLE OF CONTENTS

PAGE

Page

LIST OF ILLUSTRATIONS

Page

Page

LIST OF TABLES

CHAPTER 1

INTRODUCTION AND LITERATURE REVIEW

CHAPTER 1

INTRODUCTION AND LITERATURE REVIEW

Simulators play an increasingly more important role in enhancing the training of pilots, nuclear plant operators, chemists, engineers, astronauts, and supertanker captains to name just a few. Simulators are valuable as they may be used to teach and reinforce a set of reflexes for the routine operation of complex equipment without risk to human life if errors are made. In addition the underlying principles of a complicated process can be demonstrated with a simulator in ways that would be impossible or difficult to show while learning takes place on a real system.

Potentially, anaesthesiology should benefit from simulator technology. The equipment used is complicated, anaesthetic agents are amongst the most potent and fastest acting of drugs, and the anaesthetist must be able to act quickly and decisively in a moment of crisis. During training the anaesthetist learns theoretical skills from text books and lectures, but still has to acquire practical skills, under supervision, in the operating theatre where there is limited opportunity to experiment.

One aspect of anaesthetic training that may be improved by simulation techniques is the teaching of the uptake and distribution of a volatile anaesthetic agent during anaesthesia. This is a difficult process to visualise and would be easier to understand if it was possible to show the concentration of anaesthetic agent in various tissues in real

time graphical form.

This dissertation covers aspects of the design of a simulator that can be used to enhance a students understanding of the uptake and distribution of the volatile anaesthetic agent halothane. The simulator consists of two sections : a model of the human cardiovascular and respiratory systems (including the anaesthetic aperatus) and the computer software that allows a student to perform a simulated anaesthetic procedure while plotting graphs of various model parameters.

The following sections cover basic anaesthesia and anaesthetics related equipment and is based on the following works : Atkinson et al (1987), Davison (1968), Stoelting and Millar (1984). In addition a literature review of simulation and modelling in connection with anaesthesiology is presented.

1.1 - HISTORY OF INHALATION ANAESTHETIC AGENTS.

The use of inhaled agents for the purpose of providing anaesthesia has been in regular practice for the last 140 years. Ether has been used since 1846, chloroform since 1847, and nitrous oxide since 1862. Up until the 1950's all volatile anaesthetics were either explosive in oxygen, toxic, or both. The development of fluorine technology during the 2nd World war led to the discovery and clinical use of a new generation of non-explosive volatile anaesthetic agents : halothane in 1956, methoxyflurane in 1959, enflurane in 1972, and isoflurane in 1981 . The latest

anaesthetic agent soon to be made available is the long awaited sevoflorane that was identified in 1972, and has since then undergone extensive testing (Holaday and Smith 1981)

In the early days of anaesthesia, the apparatus required for the delivery of an anaesthetic agent to a patient was very simple, and consisted of an ether or chloroform soaked piece of towel that was placed over the patients mouth and nose. Later improvements included various designs of gauze-covered masks (such as Skinners wireframe mask and the Shimmelbusch mask) that were placed over the patients nose and mouth while ether was dripped onto it (Secher 1982).

Recent technological advances have introduceda large range of complicated anaesthetic related equipment. Anaesthetic and respiratory gases (a mixture of oxygen, nitrous oxide, and either halothane, or isoflurane) are delivered to the patient via a maze of pipes, valves, and connectors. In addition to gas delivery systems the patient's physiological status is monitored with a number of different electronic or mechanical monitors, and controlled by means of a wide range of intravenous drugs.

1.2 - MODERN ANAESTHESIA.

The ideal anaesthetic agent should produce the following in a patient : sleep or amnesia, relief from pain, immobility or relaxation, and suppression of reflexes. The agent and its metabolites should not be toxic and its actions should be reversible.

All modern anaesthetic agents can produce these actions if they attain sufficient concentration in the body. However some agents are better at producing one or more of these actions than others. Furthermore they may be harmful if used in sufficient concentration to produce all four actions. Thus anaesthesia today is achieved with a combination of drugs, gases and volatile agents, each with a more selective action.

Although rendering the patient insensitive to pain during surgery is of obvious importance it is not the only activity which concerns the anaesthetist. The major activity during anaesthesia consists of maintaining the functions of the patients vital-organ systems in equilibrium by counteracting the effects of disease, surgery, and the anaesthetic agents themselves. The anaesthetist achieves this by the continuous or repeated measurement and adjustment of various physiological parameters by mechanical, pharmacological and physical means.

The following sections discuss the modern anaesthetic workplace and the procedure followed in giving a general anaesthetic today.

1.2.1 - THE ANAESTHETIC WORKPLACE.

The anaesthetic workplace under supervision of the anaesthetist consists of the following four interacting elements as shown in figure 1.1 :

1) The anaesthetic hardware
2) The physiological monitors
3) The pharmacological agents
4) The patient

The anaesthetist delivers a mixture of anaesthetic and respiratory gases to the patient via the anaesthetic hardware while constantly monitoring the patients physiological status. In addition, the patient may be artificially ventilated, and other drugs may be given intravenously to maintain the patient in a stable but reversible state of anaesthesia.

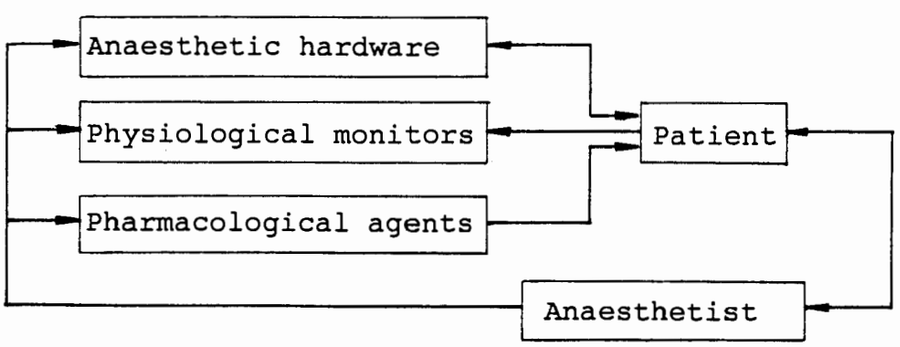


FIGURE 1.1 - The elements of anaesthesia.

1.2.1.1 THE ANAESTHETIC HARDWARE.

The anaesthetic equipment enables the anaesthetist to deliver a selectable mixture of anaesthetic and respiratory gases to a patient at a desired flow rate and tidal volume. The equipment consists of the following two main sections as shown in figure 1.2: the anaesthetic machine which controls the fresh gas flow to the patient, and the breathing circuit which connects the anaesthetic machine to the patient.

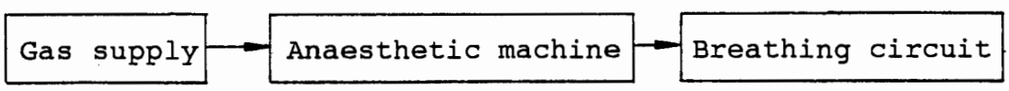


FIGURE 1.2 - Components of the anaesthetic hardware.

THE ANAESTHETIC MACHINE.

The anaesthetic machine controls the flow of anaesthetic and respiratory gases from either high pressure gas cylinders or from the hospital gas piping system to the patient and can be conveniently divided into three parts as shown in figure 1.3 :

1) The high pressure system (HPS);
2) The intermediate pressure system (IPS);
3) The low pressure system (LPS).

The high pressure system receives nitrous oxide and oxygen from high pressure gas cylinders and reduces the gas pressures to around 4 atmospheres making them more

stable. The intermediate pressure system receives nitrous oxide and oxygen from either the hospital piping system or the high pressure system outlet when piped in gas is not available. These gases then pass through a flowmeter assembly. The flowmeter assembly consists of needle valves that control the gas flow rate, followed by a flow-meter, or rotameter, that indicates the actual flow rate of that gas. The gases are then mixed and flow into the low-pressure system.

A number of safety devices are incorporated into the intermediate pressure system. Oxygen pressure is constantly monitored causing an alarm to sound if the pressure drops. In addition oxygen pressure-operated failsafe valves can stop the flow of other gases to the low pressure system in the case of an oxygen pressure failure thus avoiding the provision of hypoxic gas mixtures.

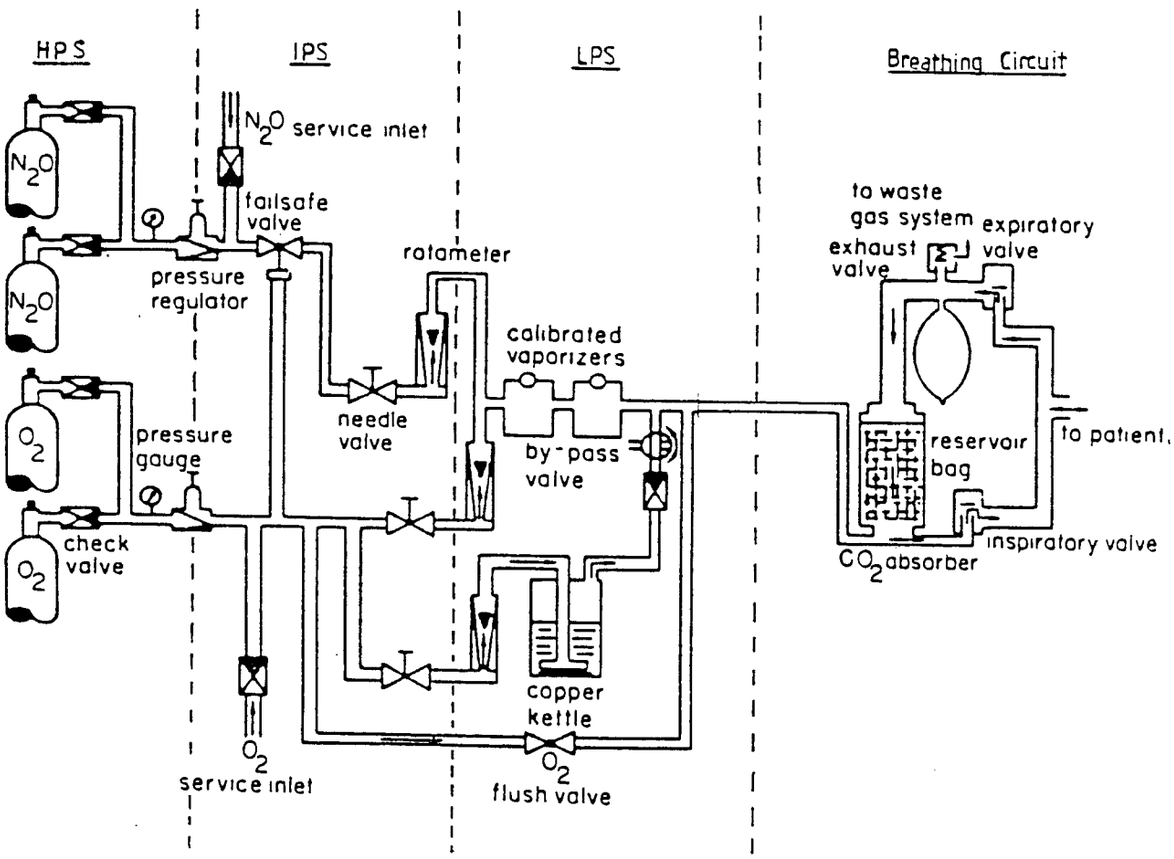


Figure 1.3 A block diagram of a typical anaesthetic delivery system. Oxygen and nitrous oxide enter the anaesthetic machine from gas cylinders attached to yokes on the machine or from a central (service inlet) supply source. Check valves prevent transfilling of gas cylinders or flow of gas into the central supply source. Pressure regulators reduce pressure in the tubing from gas cylinders to about 50psi. The failsafe valve prevents flow of nitrous oxide if the pressure in the oxygen supply circuit decreases below about 30psi. Needle valves control gas flows to flowmeters (rotameters). Agent specific vaporisers (calibrated vaporisers) provide a means to deliver a preselected concentration of volatile anaesthetic. An interlock allows only one vaporiser to be on at a time. The copper kettle is an alternative type of vaporiser used in some cases. A bypass valve must be in the ON position for effluent gas from the copper kettle to enter the total gas flow. Any flow metered to the copper kettle when the bypass valve is in the OFF position is vented to the atmosphere. After mixing in the manifold of the anaesthetic machine, the total gas flow enters the breathing system (in this case a circle system) where unidirectional valves assure gas from the patient flows through the carbon dioxide absorber. Excess gases are vented through the overflow valve (exhaust valve) into the gas scavenging system (waste gas system). The gas reservoir bag compensates for variations in inspiratory flow rate. (Reproduced from Stoelting and Miller 1984).

An Oxygen flush valve is also normally provided to allow the anaesthetist, in an emergency, to flood the breathing circuit with a high flow of pure oxygen.

The low pressure system is the part of the machine downstream of the flowmeters in which the combined gas pressure is slightly above atmospheric. The components normally found in the low pressure system are the following : vaporisers, vaporiser circuit control valves, back pressure safety devices, and the common gas outlet. As most general anaesthetic agents in use today are liquids at room temperature, they must be converted into a vapour before they can be used. A vaporiser is an instrument that adds a controlled amount of vaporised anaesthetic agent (such as halothane or enflurane) to the gas (oxygen and nitrous oxide) that is flowing through it. Most anaesthesia machines have two vaporisers available thus allowing the anaesthetist to select his agents.

When a ventilator is being used for controlled ventilation of the patient, positive back pressure from the breathing circuit during the inhalation cycle can cause the concentration of vaporized anaesthetic agent issuing from the vaporizer to be higher than that set on its dial. High pressures in the low pressure system can also increase leaks and cause inaccurate flow-meter readings. Thus most machines that utilize a ventilator in the breathing circuit will have a back pressure safety device fitted between the low-pressure system and the breathing circuit.

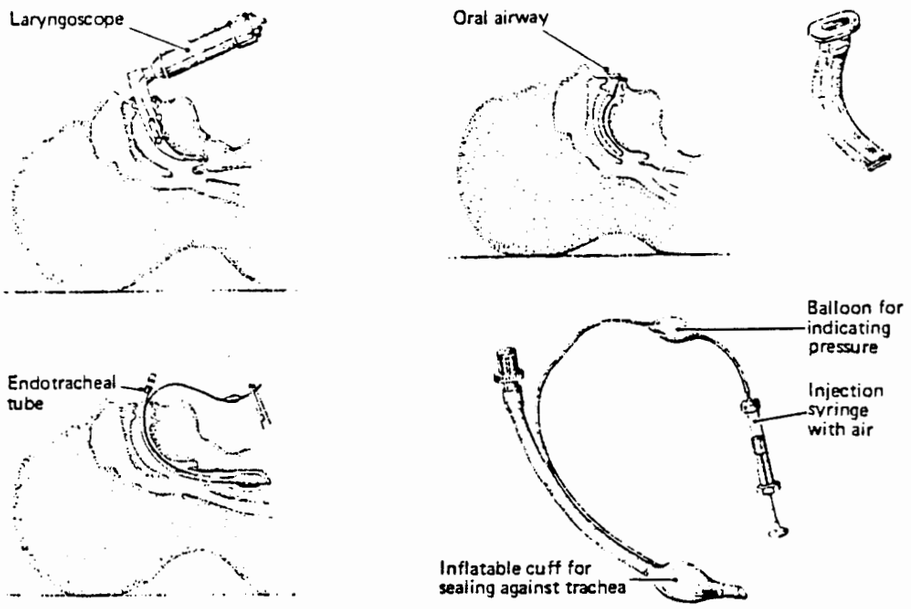


Figure 1.4 Breathing circuit attachment to the patient. (Reproduced from Jacobson and Webster 1977).

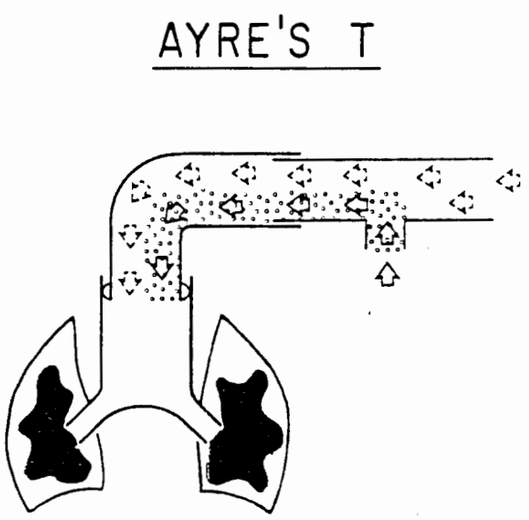


Figure 1.5 Open breathing circuit : Ayres T piece. Inflowing anaesthetic vapour is shown as arrows with continuous lines and ambient air by arrows with dashed lines. In the diagram Ayres T is connected to an *in situ* endotracheal tube. (Reproduced from Eger 1974).

THE BREATHING CIRCUIT.

The breathing circuit consists of a set of components that provides the following functions :

1) Delivery of respiratory and anaesthetic gases to the patient at predictable flow rates and concentrations.
2) The elimination of expired carbon dioxide.
3) The control and monitoring of the respiratory function of the patient.

The breathing circuit receives the gases from the low-pressure system of the anaesthesia machine and is connected to the patient usually by means of a face-mask or endotracheal tube as demonstrated in figure 1.4. An oral airway can be used with the facemask to keep the airway open and to protect the tongue from biting.

Anaesthetic breathing circuits are classified as open, semi-open, semi-closed, and closed according to the presence or absence of 1) a gas reservoir bag, 2) rebreathing of exhaled gases, 3) carbon dioxide absorbtion, and 4) unidirectional valves. Table 1.1 lists the various breathing circuits and indicates the required fresh gas inflow rate for adequate anaesthesia.

System	Gas bag	Rebreathing	CO_2 Removal	Uni-Dir Valve	Fresh gas inflow rate[b]
OPEN					
Insufflation	No	No	No	None	Unknown
Open drop	No	No	No	None	Unknown
Ayres T	No	No	No	None	Unknown
SEMI-OPEN					
Mapleson A,B C, and D.	Yes	No[a]	No	One	High
Bain	Yes	No[a]	No	One	High
Mapleson E	No	No[a]	No	One	High
Jackson-Rees	Yes	No[a]	No	One	High
SEMI-CLOSED					
Circle	Yes	Partial	Yes	Three	Moderate
CLOSED	Yes	Total	Yes	Three	Low

[a]No rebreathing only when fresh gas inflow adequate.
[b]High = > 6L/Min, Moderate = 3-6 L/Min, Low = 0.3-0.5 L/Min.

TABLE 1.1 Classification of anaesthetic breathing systems. (After Stoelting and Miller 1984 Table 11.5.)

i) OPEN BREATHING CIRCUITS.

Circuits such as the open drop, insufflation, and Ayres T piece are not often used today. The Rees' modification of Ayres T piece is still used in modern paediatrics. Open systems are very simple, but have a few disadvantages. They need a high fresh gas flow and are thus wasteful of anaesthetic gas and vapour, they pollute the operating theatre enviroment, and the lack of a reservoir prevents assisted respiration. Their greatest defect is that they produce an unstable anaesthetic state (Eger 1974).

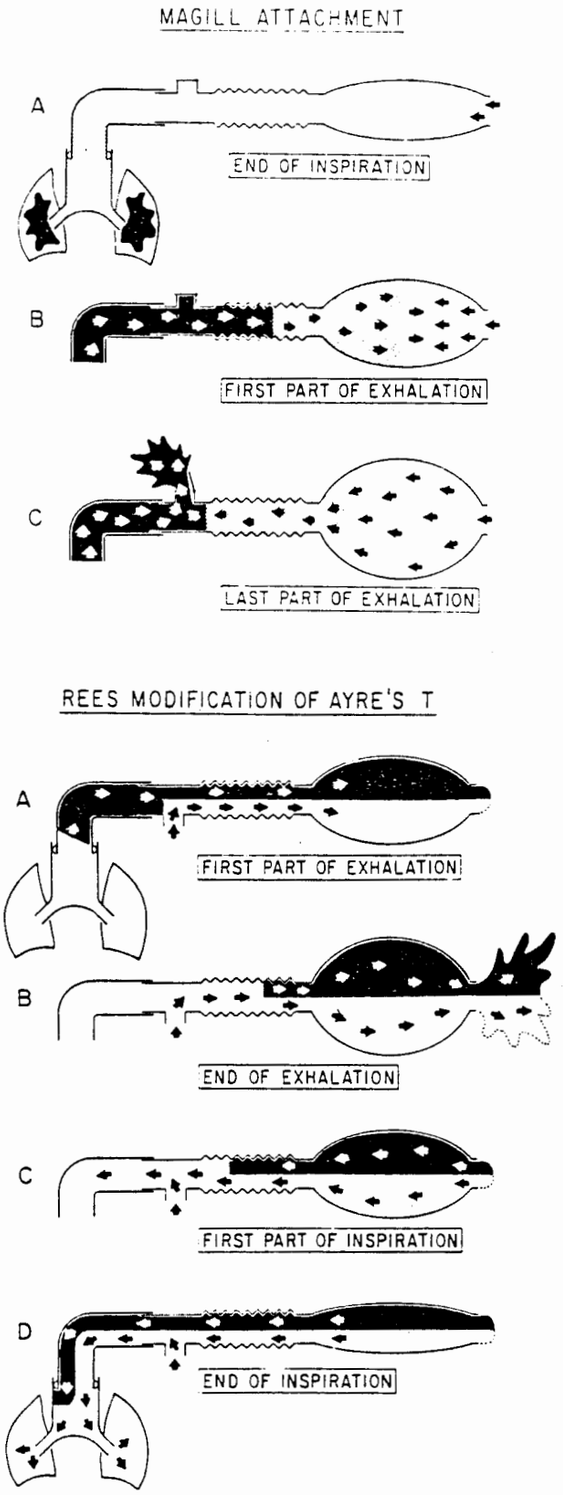


Figure 1.6 Semi-open breathing circuits : The Magill attachment and the Rees modification of Ayres T. The Rees modification adds to the "T" a tube reservoir plus a rebreathing bag which also acts as a reservoir. The black area with white arrows represents alveolar gas (which contains CO_2 and depleted anaesthetic vapour) and the clear area with black arrows represents fresh gas (O_2, N_2O, and anaesthetic vapour). The stippled area represents dead space gas. The devices may be connected to a mask or to an endotracheal tube (Reproduced from Eger 1974).

ii) SEMI-OPEN BREATHING CIRCUITS.

These systems are generally open systems that have had a reservoir and possibly one or more valves added to them (figure 1.6). They allow the effective use of less potent agents such as nitrous oxide. Respiration may be assesed from the movement of the resevoir bag, and compression of the bag may be used to assist or control respiration. They provide a more stable anaesthetic state; ie, an increase in ventilation rate is less likely to dilute the inspired gas with ambient air, nor can a decrease in ventilation rate raise the inspired concentration as may occur with open-systems. However, these systems are wasteful of anaesthetic gas and cause excessive heat loss and drying of the respiratory mucosa. Rebreathing will occur (with a corresponding rise in inspired carbon dioxide concentration) if gas delivery rates are too low (Eger 1974).

iii) SEMI-CLOSED AND CLOSED BREATHING CIRCUITS.

Absorption of carbon dioxide from the breathing circuit gives these systems an advantage over the previously discussed systems. It permits the use of low gas delivery rates and rebreathing without elevating inspired carbon dioxide. In addition low gas delivery rates reduce the contamination of the operating theatre by anaesthetic gases. Finally, rebreathing humidifies the inspired gas and thereby reduces the drying of the respiratory mucosa. The two main types of semi-closed or closed

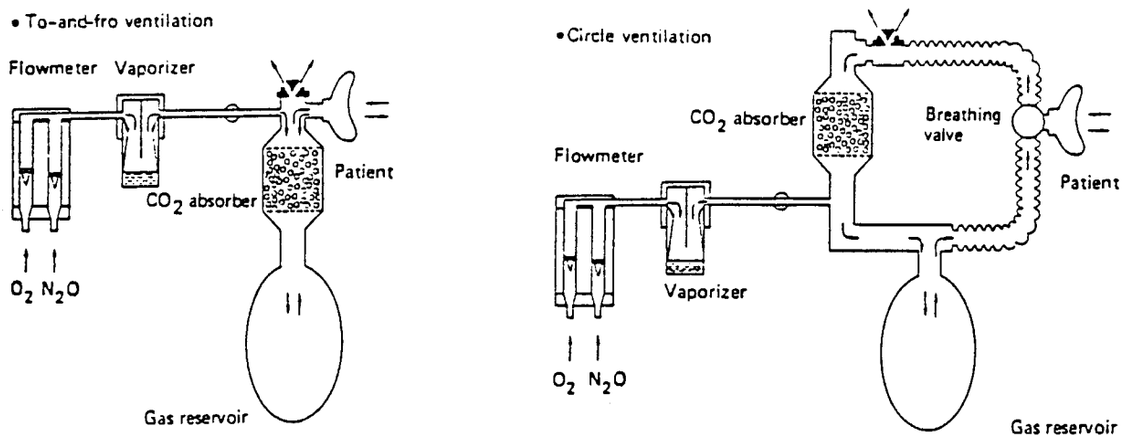


Figure 1.7 Semi-open breathing circuits : Partial rebreathing. (Reproduced from Jacobson and Webster 1977).

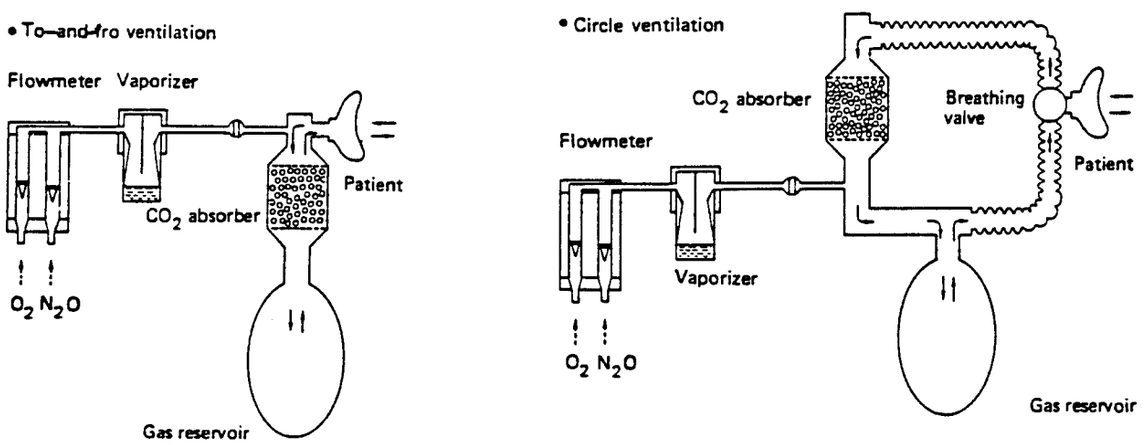


Figure 1.8 Closed breathing circuit : Full rebreathing. (Reproduced from Jacobson and Webster 1977).

breathing circuits are the so called To-and-Fro system and the circle absorption system as demonstrated in figures 1.7 and 1.8. The To-and-Fro system is not very popular today, making the circle absorbtion system the most widely used modern breathing circuit system (Eger 1974).

Most major surgical procedures require the patient to have a degree of muscle relaxation varying from mild flacidity to total paralysis. This is achieved through the use of muscle relaxant drugs which affect the ability of the patient to breathe spontaneously. Thus a ventilator of some sort is normally included in the breathing circuit to provide assisted ventilation when required. Patients who are being ventilated are normally attached to the breathing circuit by means of an endotracheal tube. This bypasses the upper respiratory tract which is responsible for the warming and moistening of inspired air. Thus a humidifier or nebulizer is often included in the breathing circuit. A temperature sensor in the patient's airway provides feedback for the heater in the humidifier so as to allow it to keep the inspired gas saturated with water vapour at body temperature.

A number of monitoring aids are commonly used to warn of mechanical malfunctions in the breathing circuit. An oxygen analyzer can be used to continuously monitor and indicate the concentration of oxygen in the inspired air. If a ventilator is used a low, and less frequently, a high pressure alarm can be incorporated. These indicate if the breathing circuit has become disconnected from the patient (low pressure) or if excessively high airway pressures are

being generated. A spirometer is used to measure the patient's tidal volume.

1.2.1.2 THE PHYSIOLOGICAL MONITORS.

The purpose of monitoring is to obtain information about the patient that will A) indicate to the anaesthetist whether physiological homeostasis is being maintained, B) alert the anaesthetist of undesirable changes so that therapeutic intervention may take place, and C) allow the anaesthetist to assess the result of this intervention.

A large amount of information may be obtained about the patient's status by means of simple observation, auscultation, and palpation. However, the patient is often covered with sterile drapes, and may be so positioned as to make it difficult for the anaesthetist to perform these observations. Thus a number of mechanical and electronic monitoring aids are used. The degree of monitoring required depends on the severity of the surgical procedure and on the patients physiological condition. Some of the more commonly used monitors are the electrocardiograph (ECG), peripheral pulse monitor, core temperature monitor, capnograph, pulse oximeter, arterial and venous blood pressure monitors, respirometer, and urine monitors. Some monitors are electronic and display a waveform on a screen, while others are mechanical in nature and the parameter is indicated by means of a dial, column of water, or digital display.

1.2.1.3 PHARMACOLOGICAL AGENTS.

The modern anaesthetist is required to have a detailed knowledge of the actions and side effects of the anaesthetic drugs that he uses. In addition he must know the actions and side effects of the drugs used by his surgical and medical colleagues as they may have a marked effect on the course of anaesthesia. These agents can be catagorised according to their main actions and the purposes for which they are used (Vickers et al 1984) :

1) Central nervous system depressants.
2) General anaesthetic agents.
3) Analgesic agents.
4) CNS stimulants.
5) Neuromuscular blocking agents.
6) Parasympathetic and cholinergic agents.
7) Parasympathetic antagonists and anticholinergic agents.
8) Cardiovascular agents.
9) Other agents.

1.2.1.4 THE PATIENT.

Finally it is important that the anaesthetist considers the status of the patient. The patient may be young or old, and may be in a stable or a critical condition. The anaesthetist is required to choose the type of anaesthesia appropriate to the surgical procedure that is to be performed and the position of the patient on the operating table. The patient responds to anaesthesia and surgery according to his underlying physiological status

which the anaesthetist monitors via the monitoring equipment, as well as by clinical signs such as pupil diameter, skin colour, and reflex movements. It is the anaesthetists duty to maintain the equilibrium of the patient's vital organ systems during surgery, while keeping the patient relaxed, pain free and (in the case of general anaesthesia) totally insensible to his surroundings.

1.2.2 THE GENERAL ANAESTHETIC PROCEDURE.

This section discusses the most commonly used techniques for performing inhalation anaesthesia, using the previously discussed equipment and drugs. All of these techniques proceed in the following 4 stages:

1) Pre-operative assesment and medication;
2) Induction of anaesthesia;
3) Maintenance of anaesthesia; and
4) Reversal and recovery from anaesthesia.

These stages are discussed in more detail in the following sections.

1.2.2.1 PRE-OPERATIVE ASSESMENT AND MEDICATION.

Unless the surgery that is to be performed is an emergency, the anaesthetist will visit the patient the day before the operation to assess the patient's fitness for anaesthesia, to prescribe drugs for pre-medication and find out what drugs the patient may already be taking, and to discuss the pending procedure with the patient.

During this time the anaesthetist will decide on the anaesthetic technique that is most appropriate for the type of surgery to be performed.

The object of pre-medication is :

1) To allay anxiety and fear.
2) To reduce respiratory tract secretions.
3) To enhance the hypnotic effect of general anaesthetic agents.
4) To reduce post-operative nausea and vomiting.
5) To reduce Vagal reflexes caused by intubation.

1.2.2.2 INDUCTION OF ANAESTHESIA.

Before the patient is brought into theatre, the anaesthetist makes a thorough check of all his equipment, and makes sure that all the required equipment and drugs are in their proper places and available to him. The patient is then brought into theatre, placed on the operating table, and connected to the monitoring devices prior to induction of anaesthesia. Two main methods of induction are generally used: inhalation induction and intravenous (IV) induction.

INHALATION INDUCTION.

This is a relatively slow method of induction that is used with young children, patients with airway obstruction, or poor risk patients unsuitable for IV induction. The patient is allowed to breathe a mixture of anaesthetic

gases and oxygen through a facial mask until unconsciousness is achieved. During induction the patient's repiratory pattern, skin colour, pulse, blood pressure, and electrocardiograph (ECG) waveforms are constantly monitored in case of any adverse reactions of the patient to the anaesthetic.

INTRAVENOUS INDUCTION.

This is a fast method of induction that is achieved by introducing an anaesthetic agent directly into the blood stream. Firstly, intravenous access is established in one of the patient's subcutaneous veins. The patient is then allowed to breathe pure oxygen through a face mask as a slow injection of the intravenous agent is given. As these agents are short acting, anaesthesia can only be maintained, and deepened, by rapid introduction of inhalation anaesthetic agents to the patient's inspired air, by repeated bolus injections of the intravenous agent, or by continuous intravenous infusion of the agent.

1.2.2.3 MAINTENENCE OF ANAESTHESIA.

Once the patient has lost consciousness it is the duty of the anaesthetist to bring the patient to the desired level of anaesthesia, and to maintain this for the duration of the surgical procedure. The patient is allowed to breathe a mixture of respiratory and anaesthetic gases (Typically 1/2% atmospheric pressure halothane in a 70% nitrous oxide and a 30% oxygen gas mixture). The concentration of the anaesthetic agent is set according to

its potency. A potent anaesthetic agent is one which causes anaesthesia at low concentrations, while a less potent agent requires higher concentrations.

THE POTENCY OF AN ANAESTHETIC AGENT.

The minimum alveolar concentration (or MAC) of a volatile agent is the alveolar concentration required of that agent which will prevent reflex movement in response to surgery in 50% of patients and can be used as a measure of the potency of the agent (Eger et al 1965). The value of MAC for each agent is known to be fairly constant although it may change significantly with the patient's age and body temperature. The presence of narcotics, opiates, alchohol, sedatives, and central nervous system catecholamines in the patient can also cause changes in MAC (Eger 1974). The effect on MAC of several agents given simultaneously is additive, ie 1/2 MAC N_2O plus 1/2 MAC halothane is equivalent to a total dose of 1 MAC.

MAC values for some commonly used anaesthetic agents are shown in table 1.1.

AGENT	MAC (% atm)
Halothane	0.75
Isoflurane	1.15
Enflurane	1.68
Nitrous oxide	105
Sevoflurane	1.71*

TABLE 1.2 - MAC values of common anaesthetic agents. (Adapted from EGER 1974, *Holaday and Smith 1981)

The time taken for the alveolar concentration of an anaesthetic agent to equilibrate with the inspired concentration is dependent on certain factors such as the solubility of the agent, ventilation rate, and the volume of the breathing circuit. Thus the inspired concentration should be greater than the MAC. To eliminate movement in 95% of patients requires an alveolar concentration which exceeds MAC by 10 to 40 %.

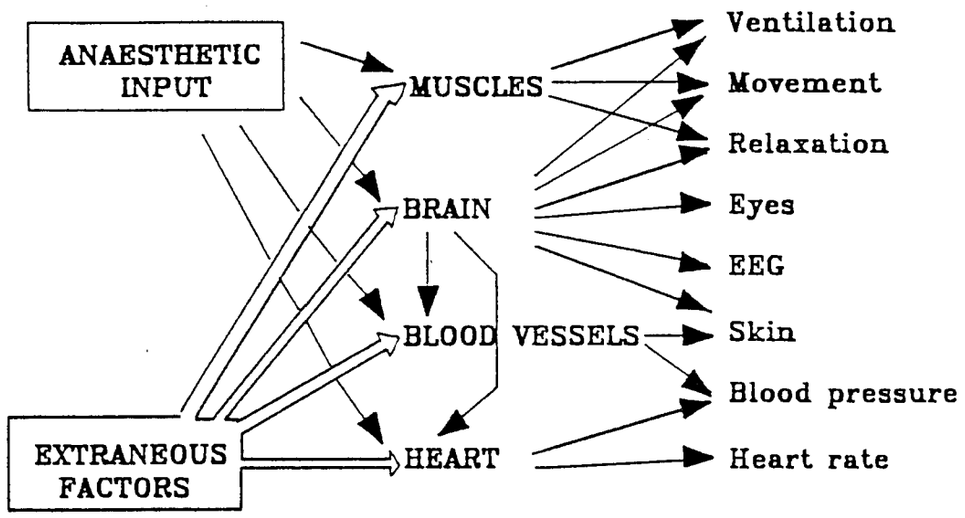


Figure 1.9 The effects of anaesthetic agents and their indication. Anaesthetic agents can act directly on the four sites in the body illustrated, or indirectly by affecting the brains control over these areas. In addition extraneous factors such as surgery, disease, and other drugs can also act at these sites. The effect of these actions is indicated by changes in the list on the right of the diagram. (Adapted from Atkinson 1987)

Anaesthetic doses are thus expressed as multiples of MAC and they can be used to calculate the required concentration of a specific agent required to produce a desired level of anaesthesia. The anaesthetist must be aware of the factors governing the uptake and distribution of the anaesthetic agent by the patient, so that adequate anaesthesia may be maintained while avoiding excessively deep anaesthesia due to overdosage.

Figure 1.9 shows a schematic diagram that illustrates the areas that are affected by an anaesthetic agent and the interactions that can occur. Extraneous factors and the anaesthetic agent itself can cause changes in the distribution and uptake of anaesthetic vapour which will affect the time taken for the correct depth of anaesthesia to be achieved. The following sections describe the various methods of maintaining anaesthesia in greater detail.

VENTILATION DURING ANAESTHESIA.

Hypnosis, relaxation, and analgesia during anaesthesia may be supplied by a mixture of seperate agents. This allows for a lighter state of anaesthesia, with preservation of autonomic reflexes but still maintaining the patient in a relaxed and pain free state during surgery. Relaxation and analgesia is achieved by intravenous infusion of appropriate drugs in conjunction with the anaesthetic agents.

During minor surgical procedures that produce little reflex or pain stimulation the level of anaesthesia required will not affect the patient's ability to breathe and he may be allowed to breathe unassisted. The patient then breathes through a face mask with or without an oropharangeal airway in place.

If the patient is placed in such a way as to impede spontaneous ventilation or when a major surgical procedure requires the depth of anaesthesia to be such that spontaneous ventilation is affected, then the patient has an endotracheal tube placed in the upper airway after

induction of anaesthesia. The administration of a short acting depolarising muscle relaxant such as suxamethonium chloride facilitates the placement of the tube, which is then connected to the breathing circuit. The patient is still able to breathe unassisted but may recieve assistance from a mechanical ventilator connected to the breathing circuit.

Certain major surgical procedures such as intraperitoneal, thoracic, intracraneal, and prolonged surgery require marked muscle relaxation. This may be provided by intravenously injecting a long acting non-depolarising muscle relaxant such as pancuronium. These agents affect the patients ability to breathe and in these cases assisted ventilation is required.

1.2.2.4 RECOVERY FROM ANAESTHESIA.

Ideally the patient should regain consciousness as quickly as possible after surgery has been completed. To achieve this the anaesthetist estimates the time that the procedure is going to take and then calculates his drug dosages so as to maintain anaesthesia for that time. Once the procedure is completed the following steps are followed. If muscle relaxation has been used, any residual neuromuscular blockade is antagonised with a drug such as neostigmine , and spontaneous respiration is re-established. The patient may also be allowed to breathe 5% carbon dioxide to restore normocapnia as blood carbon dioxide concentrations can fall below normal when a patient is being artificially ventilated. The patient is then placed in the lateral recovery position, and the

endotracheal tube is removed. The patients ability to breathe unassisted is assesed, and the patient is allowed to breathe oxygen enriched air through a face mask. The patient is then transferred to a trolly and placed in a recovery room where he is kept under observation until fully recovered.

1.3 TRAINING IN ANAESTHESIA.

In South Africa anaesthetics training is minimal at the undergraduate level. It consists of a 2 week course in the fourth year of the medical degree which is divided up into tutorials and some theatre exposure. During housemanship the prospective anaesthetist is required to perform 40 supervised anaesthetic procedures in order to comply with the South African Medical and Dental council regulations for registration as a medical practitioner. A further 4 years of post-graduate study may then be undertaken to specialise in anaesthesia.

1.3.1 - COMPLICATIONS, MECHANICAL FAULTS AND HUMAN ERROR

Mishaps in anaesthesia occasionally occur no matter how experienced the anaesthetist may be. The anaesthetic equipment may fail, the patient may have a rare reaction to one of the drugs being used, or the anaesthetist may make an error. Serious complications can lead to the death or permanent disability of the patient.

Dripps et al (1961) reported that in a series of 80 deaths due to anaesthesia, 87% were caused by human error. Similarly, Clifton and Hotten (1963) report that in a study of 52 anaesthetic related deaths, 65% were due to human error. In a survey of anaesthetic related death at Groote Schuur Hospital, Harrison (1968) considered 58% of the deaths as probably preventable and 93% as possibly preventable. These studies have concentrated on quantifying the overall anaesthetic risk using mortality

as the only criteria for negative outcomes. However a study using a methodology known as "critical incident analysis" has been undertaken by Cooper et al (1978) to study preventable anaesthetic mishaps. A mishap was labeled a critical incident if it could have led to or did lead to an undesirable outcome such as death or disability. The incidents ranged in seriousness from laryngoscope malfunctions, with no known consequences to the patient, to breathing circuit disconnections that resulted in the death of the patient. They found that 82% of the preventable incidents reported were due to human error, and 14 % due to equipment failure. The types of human error and of equipment failure that occurred are shown in figures 1.10 and 1.11.

Wyant (1978) has produced a comprehensive text on complications caused by mechanical failures of anaesthetic equipment. Due to the complexity of modern anaesthetic equipment it would be tedious to mention here all the components of these systems that can fail. However, the major types of mechanical malfunction appear to consist of breathing circuit disconnections or blockages, laryngoscope malfunctions, and electronic monitor failures.

Patient complications may occur during an operation that can not be prevented by the anaesthetist but may be corrected. The patient may develop complications such as bronchospasm, hypotension, hypertension, cardiac arrhythmia, embolism, or hypothermia. The patient may be difficult to intubate due to congenital, anatomical or acquired causes. Fell (1985) indicates that one in sixty

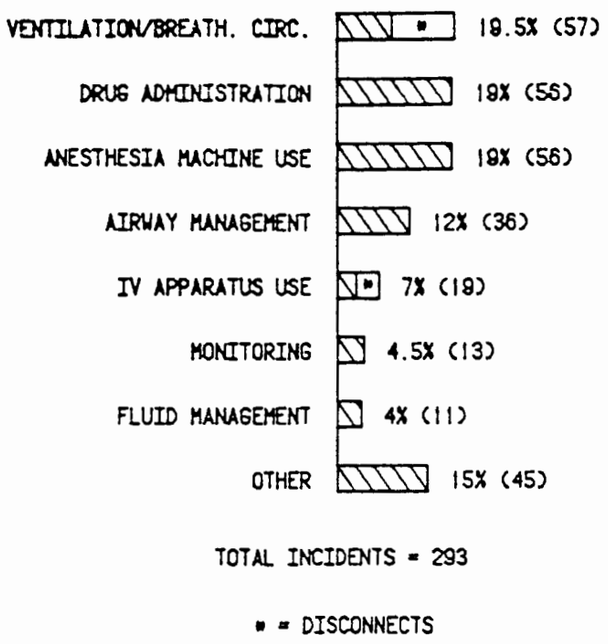


Figure 1.10 Distribution of human errors. The numbers in parentheses represent the actual number of incidents that were reported. (Reproduced from Cooper et al 1978).

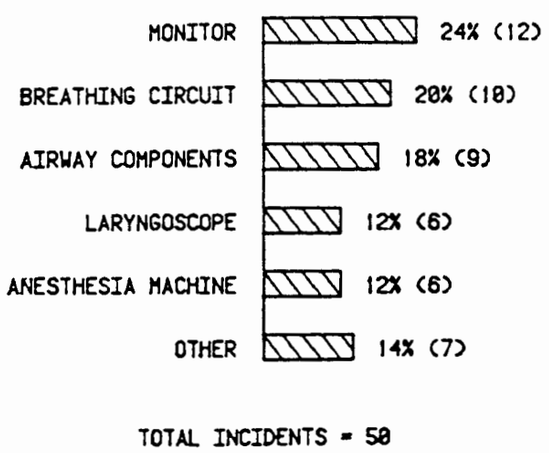


Figure 1.11 Distribution of equipment failures. (Reproduced from Cooper et al 1978)

five patients is likely to present difficulties in tracheal intubation. Another complication that sometimes arises is patient reaction to drugs and infused blood products.

In 1984 a study was undertaken to determine the stress an anaesthetist undergoes while giving an anaesthetic and to evaluate whether training and experience would have any effect on this response (Toung et al 1984). Heart rate was selected as an indicator of stress and ECG studies were performed on 1st and 2nd year trainees as well as on resident anaesthetists. The results of the study are shown in figure 1.12.

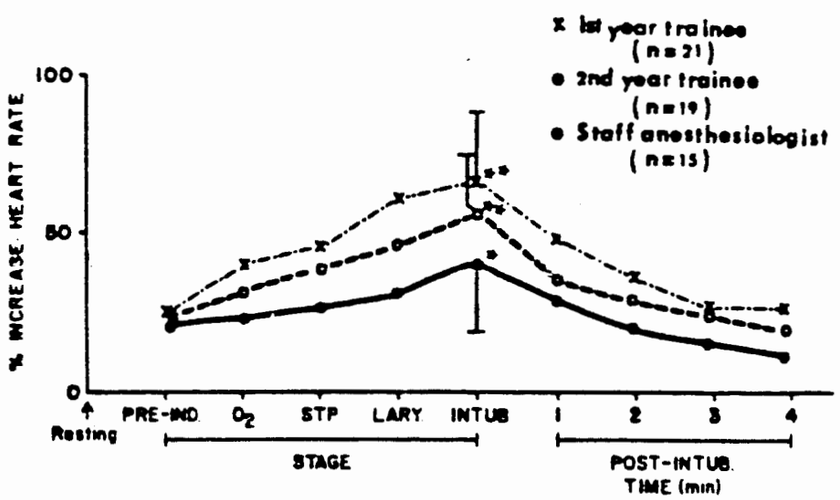


Figure 1.12 Heart rate as a measure of the stress of giving an anaesthetic. Changes in the heart rates of 3 groups are shown during and after induction of general anaesthesia. (Reproduced from Toung et al 1984).

Toung found induction of anaesthesia to be the most stressful part of the procedure, but that this stress was lessened by training and experience.

The following section consists of a review of simulators that have been created to supplement training in anaesthesia.

1.3.2 - TRAINING AIDS

A computer controlled manikin, designed specifically for the practical training of anaesthetic interns in the art of intubation and anaesthetic induction has been developed. (Denson and Abrahamson 1969). This simulator, known as Sim One, consists of a manikin connected to an analogue computer and operator console. The manikin looks like a real patient, breathes with its chest and abdomen, has carotid and temporal pulses synchronous with an audible heartbeat, and can be ventilated by bag and mask or through an endotracheal airway. The "patient" also responds appropriately when oxygen, nitrous oxide, thiopental sodium, succynylcholine, and either of two vasopressors are given. The computer monitors the actions of the student and the responses of the patient and produces a printed record at the end of the procedure.

The assumed advantages of this simulator are as follows: " The use of the simulator allows for a planned and gradual increase in the difficulty of the problems to be solved by the anaesthetist resident instead of his performing new tasks in the operating room as the necessity for them arises. Second, the use of a simulator permits almost unlimited repetition of any phase of the procedures to be learned. Third, the trainee can obtain immediate feedback on his own performance. Finally, the resident can proceed at his own rate" (Abrahamson et al 1969).

This simulator was evaluated for its educational potential in training residents in the art of intubation (Abrahamson et al 1969). It was found that residents who used the simulator prior to trying to intubate a real patient achieved proficiency levels in a smaller number of training days, and in a smaller number of trials in the operating room than those who did not use the simulator.

Sakurai et al (1980) describe a patient robot system for resuscitation training. The simulator consists of a manikin that has sensors that detect the actions of external cardiac massage, artificial respiration and the injection of drugs. A microcomputer processes the sensory inputs. Parameters such as heart rate, arterial pulsations, breathing, light reflex of the pupil and blood pressure are displayed on a TV screen. The software evaluates the procedure and indicates the effectiveness of the students actions.

Hon (1982) describes a similar system for cardiopulmonary resuscitation training. However this system is connected to an interactive videodisk instruction setup. The student is asked questions by the system via a TV screen, and answers using a light pen or by performing actions on a life sized manikin. The videodisk system can show video clips of how the procedure should be carried out, as well as still pictures and computer graphics that indicate how effectively the student is performing. The simulator can be used by students without the help of an instructor. The interactive system can handle almost any difficulty a student may experience and gives the required response to enable the simulation to proceed.

1.3.3 MODELING IN ANAESTHESIA.

A number of mathematical models of the uptake and distribution of anaesthetic gases have been developed. Most of the earlier models developed are based on linear system equations. This implies that the system under study remains unaltered in the presence of the anaesthetic drug. ie. halothane is assumed to behave like an inert gas and to have no effect on the cardiovascular system or the respiratory system of the model.

In 1950 Kety published a physical description of the dynamics of the uptake and distribution of an inert agent in the body. In 1951 he reviewed a number of publications of mathematical representations , or models, of anaesthetic gas equilibration in the body and presented a model of his own design (Kety 1951). He modelled the body as several tissue compartments whose anaesthetic concentrations approach equilibrium with the inspired concentration. Basic assumptions that are the basis of most modern models of this sort were defined.

Mapleson (1963) developed an electrical analogue model of the uptake and distribution of an inert gas. The model is shown in figure 1.13 and consists of various tissue compartments. Each compartment is made up of a conductance and a condensor. These are respectively made proportional to the conductances and compliances of the various body compartments. If the input voltage is made proportional to the inspired concentration then the subsequent currents and voltages of the conductors and condensors will be

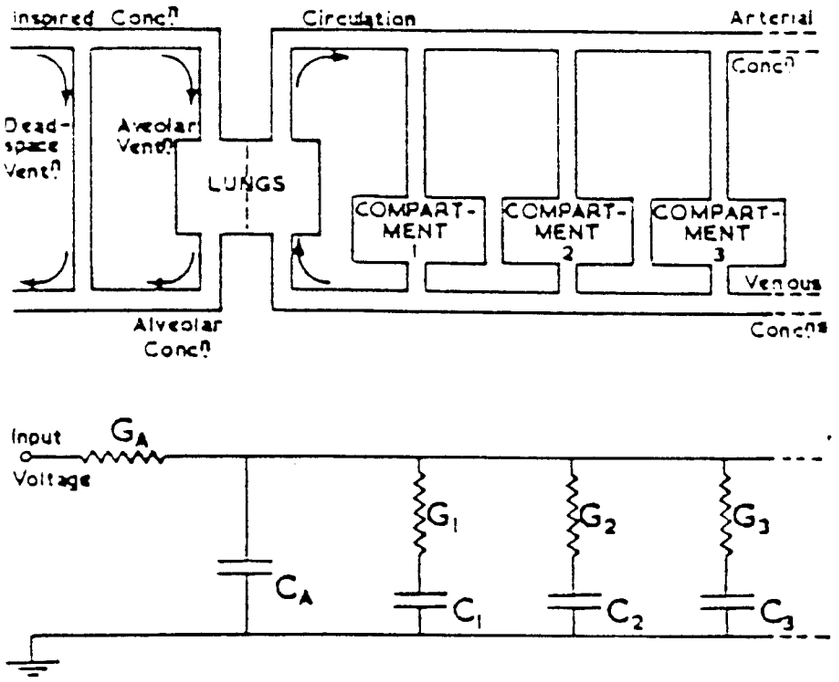


Figure 1.13 Idealised model of the body and the electrical analogue. (Reproduced from Mapleson 1963).

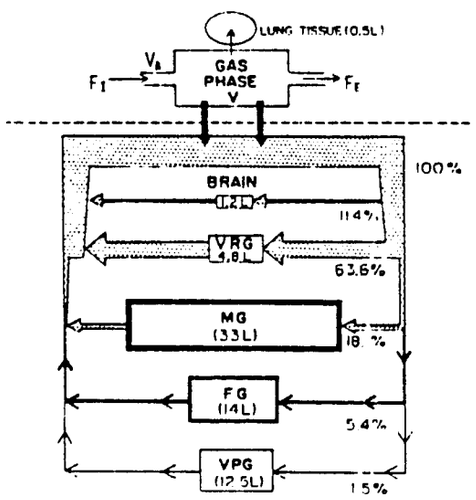


Figure 1.14 A 5 compartmental linear uptake and distribution model. The model consists of 5 compartments identified as VRG = Vessel rich group, VPG = vessel poor group, FG = fat group, MG = Muscle group. The lungs make a 6th compartment. Numbers indicate the volume of each compartment and the percentage of the cardiac output that flows through the compartment. (Reproduced from Munson and Bowers 1967).

proportional to the rates of uptake and the concentrations of the anaesthetic agent in the body compartments.

In 1963 Eger developed a mathamatical model that was used to simulate the uptake and distribution of halothane on a mainframe computer (Eger 1963a). The body was divided into 6 compartments, each of which consisted of tissue groups with similar blood-gas solubilities and blood supply per unit volume of tissue. The compartments were named as follows : The lungs, the brain, the vessel rich group (heart, hepatoportal system, kidneys, and endocrine glands), the muscle group, the fat group, and the vessel poor group (bone and cartilage). These compartments are shown in figure 1.14. Six differential equations that determine the time rate of change of the concentration of anaesthetic agent in the blood and in each compartment were derived. The model gave results that corresponded well with observed data (Eger and Guadagni 1963), and has been used since then by other investigators undertaking research in this field. The effects of hyperventilation (Munson and Bowers 1967) and changes in cardiac output (Munson et al 1968) on the rate of cerebral anaesthetic equilibration have been simulated using a similar model.

The concept of a multiple model (2 or more models whose outputs influence one another) in connection with cardiovascular research was first mentioned in 1968 (Beneken and Rideout 1968). A model of the circulation, based on lumped circuit approximations, was used for simulation studies of pulsatile pressure, flow, and volume

relationships. A second model, coupled to the first, was devised to simulate the flow and distribution of substances carried by the blood.

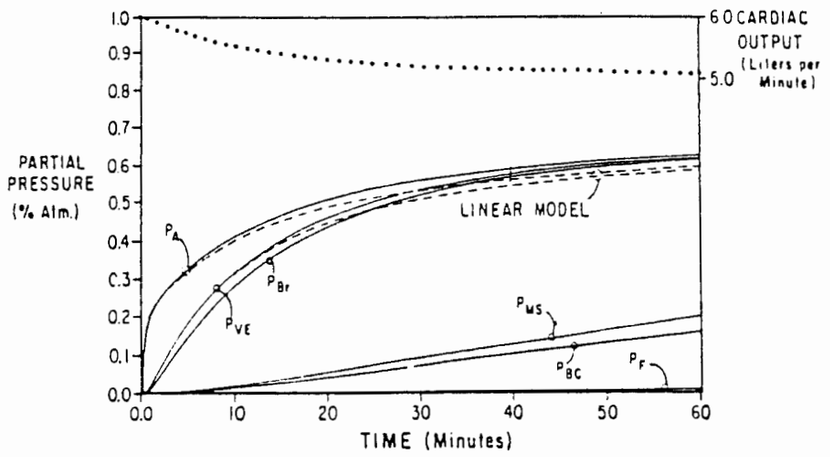


Figure 1.15 A comparison of linear and non-linear responses. Partial pressures in the non-linear model are represented by continuous lines; partial pressures in the linear model by dashed lines. Cardiac output is shown as a dotted line in the non-linear model and constant at 6 L/min in the linear one. PA = Alveolar , Pve = Viscera, Pbr = Brain partial pressure. (Reproduced from Ashman et al 1970).

A non-linear model of the uptake and distribution of halothane (Ashman et al 1970) that included the depressing effect of halothane on cardiac output was developed in 1970. Ashman felt that as anaesthetic gases have an effect on the control of respiration and circulation then these non-linear effects should also be included in the model. The model used was similar to that described by Eger (1963a) except for the equation relating cardiac output to the alveolar halothane concentration.

The equation is

$$Q = Q_0(1.0 - 0.25\ P_a) \qquad (1.1)$$

where Q_0 is the awake cardiac output and Pa is the alveolar partial pressure of halothane.

Ashman used data obtained by Eger et al (1968) to derive this relationship. On comparing the non-linear model with a model having fixed cardiac output it was found that the difference between the models was insignificant for the first five minutes of the simulation but the effect of the non-linearity was to elevate the partial pressures of halothane in the well perfused tissues and the lungs by 6% after one hour (Figure 1.15).

Up till this point in time all previous models assumed that blood flow through the various body compartments was not affected by the concentration of anaesthetic in the compartment. In 1972 a 12 compartmental analogue computer multiple model was developed that included the effect of anaesthetic gas on regional resistance to blood flow (Zwart et al 1972). The model consists of two interdependent sub-models (loops), one representing the blood circulation, and the other representing halothane transport around the body. The transport loop consists of 12 compartments : Lungs, arterial and venous blood, grey and white .matter of the brain, heart muscle, skeletal muscle, well perfused organs (kidneys, adrenal glands, and thyroid gland), poorly perfused organs (red marrow and non-fatty subcutaneous tissue), splanchnic bed (organs drained

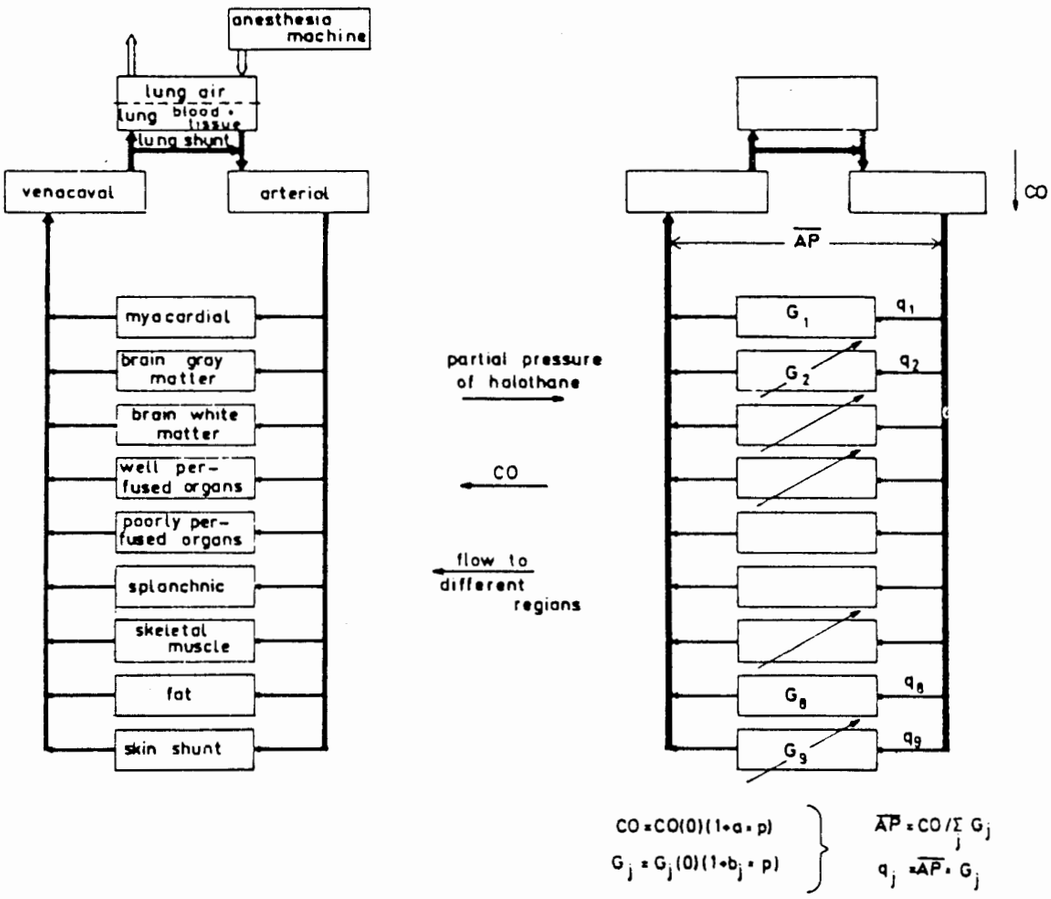


Figure 1.16 A 12 compartmental non-linear multiple model. The diagram shows a generalised scheme of the two component models (loops) which form the multiple model. *Left*, uptake and distribution loop; *right*, circulatory loop. The arrows between the loops indicate the interconnections. In the uptake and distribution loop, arterial blood is partitioned among nine parallel compartments as determined by the circulatory model. Venous blood from each compartment is then collected into the venae cavae and passes either to the lungs or through a bypass - the pulmonary shunt. In the circulatory loop, cardiac output is controlled by a selected compartmental concentration in the uptake and distribution model and divided amongst the compartments according to the conductances. Two types of conductances are used : fixed (open boxes) and variable (boxes with arrows). The variable conductances are also controlled by a predetermined compartmental concentration in the uptake and distribution model. The equations in the lower right describe the control of cardiac output (CO) and each compartmental conductance (G_j) by the selected compartmental concentration. CO(0) = cardiac output in the conscious state; a = a constant chosen such that when p = 2%, CO is decreased by 40 percent; p = concentration of halothane in volumes percent in the controlling compartment in the uptake and distribution model; G_j = conductance of the jth compartment; $G_j(0)$ = conductance of the jth compartment in the conscious state; b_j = a constant selected such that the flow in the jth compartment is that chosen to occur at 2% halothane; AP = mean arterial pressure; ΣG_j = the sum of the conductances in all the compartments (the reciprocal of this sum is the systemic vascular resistance); and q_j = the flow into the jth compartment. (Reproduced from Smith et al 1972).

by the portal and hepatic circulation), fat, and a skin shunt. Figure 1.16 shows a generalised scheme of these compartments and the circulation loop. The cardiac output and the regional (compartmental) conductances are assumed to be affected linearly in relation to the concentration of halothane in either the arterial, grey matter, or heart muscle compartments. The response of this model was found to differ significantly (7.7% after 1 hour at 1% halothane) from those generated by a standard (linear) model (Figure 1.17).

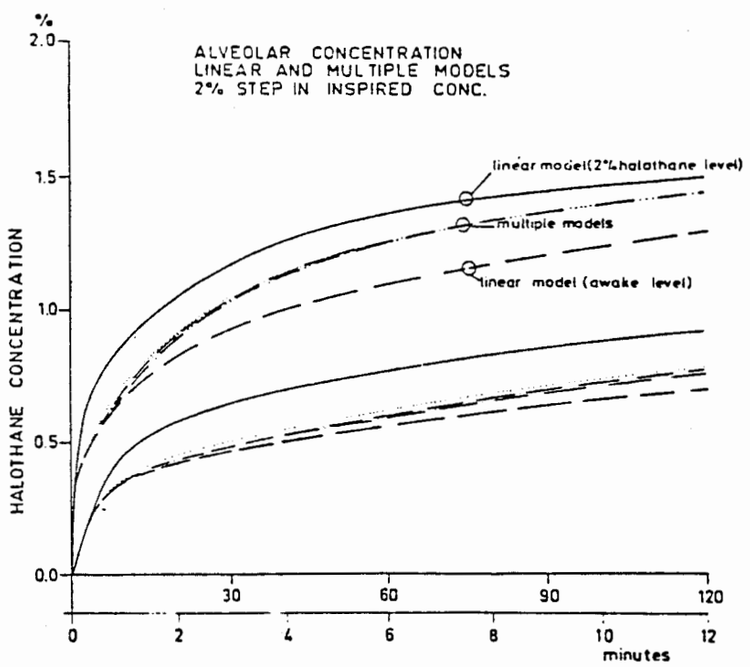


Figure 1.17 A comparison of the multiple model in linear and non-linear mode. Response of all models is to a 0 - 2% step change of inspired halothane concentration. The lower group of curves represents the first 12 minutes of the upper group. The solid line is drawn with a linear model with the flow data at 2% equilibrium level of halothane; the dashed line with flow data at awake level. The other curves were drawn with multiple models with cardiac output controlled by different compartments. .... = Arterial, -.-. = Brain grey matter, ..-..- = Myocardial control. (Reproduced from Zwart et al 1972)

In 1972 a review of computer models in cardiovascular research included a section on uptake and distribution models and featured a mathematical description of Zwarts model as an example (Beneken 1972).

In all the previously described models it was assumed that ventilation was not affected by the anaesthetic agent and remained constant during the simulation. In 1973 a model based on that of Mapleson (1963) was developed that includes the changes in ventilation and perfusion that are caused by halothane (Munson et al 1973). The effects of this agent were expressed as an empiric function of the anaesthetic partial pressure within the brain.

During spontaneous ventilation with constant cardiac output it was found that the higher the concentration of inspired gas, the slower the rate of alveolar anaesthetic concentration increase. This is due to the ventilatory depression and eventual apnoea caused by the agent, which limits the delivery of anaesthetic to the lung. Alveolar concentration did not rise above 3% of atmospheric regardless of the inspired concentration. During controlled ventilation with halothane the alveolar concentration increased as the cardiac output diminished. As cardiac arrest occured the alveolar concentration rose rapidly to approach that of the inspired concentration. Graphs that illustrate this are shown in figure 1.18a and 1.18b.

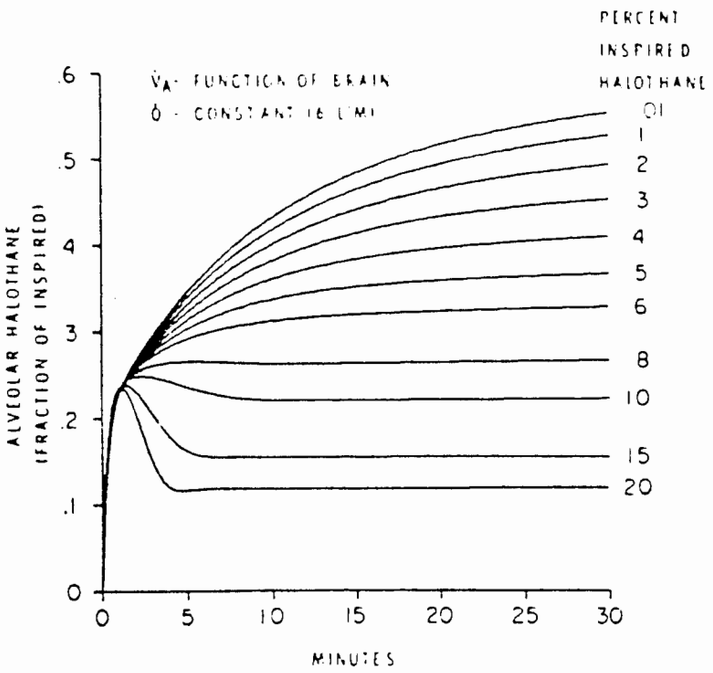


Figure 1.18a The rate of increase of alveolar concentration when ventilation is a function of the cerebral partial pressure of halothane. (Reproduced from Munson et al 1973).

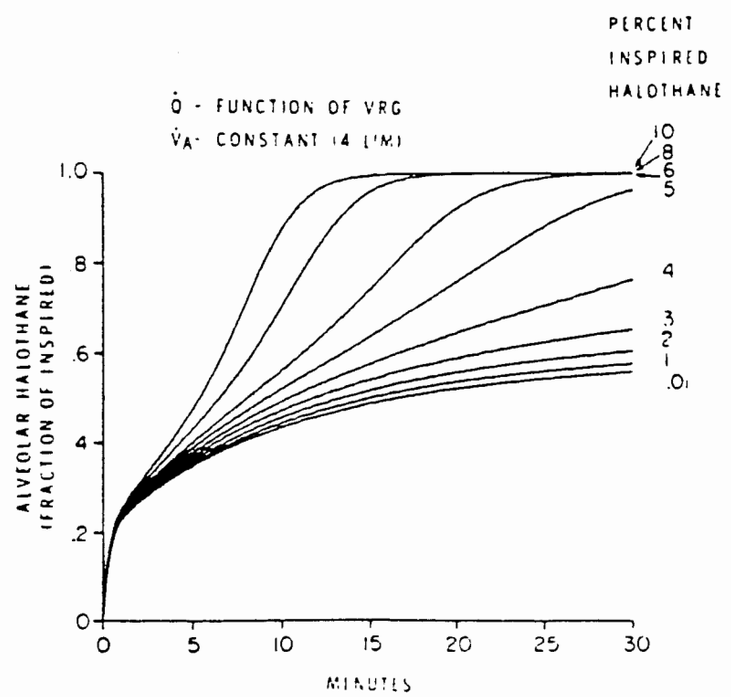


Figure 1.18b The rate of increase of alveolar concentration when cardiac output is a function of the partial pressure of halothane. Q = Cardiac output, Va = Alveolar ventilation rate. (Reproduced from Munson et al 1973).

In 1973 a review of solubility coefficients of inhaled anaesthetic agents was produced (Steward et al 1973). He compiled a table of 600 reported coefficients for 18 inhaled agents in 30 solvents. To maintain consistency all coefficients were expressed as Ostwald solubility coefficients at 37° C.

All the previously described models, with the exception of Egers model, have been electrical analogues. In 1973 a simplified way of using these models for simulation purposes using digital methods was described (Cowles et al 1973). A series of differential equations representing the model are solved by repeatedly computing a FORTRAN subroutine using a small time interval. The results are output between iterations. This method provides greater flexibility and accuracy than is available with any analogue computer. One drawback however is that it can be a slower method.

In all the previous models it has been assumed that blood flow is non-pulsatile and circulation times from lungs to tissues are negligible. In 1973, Mapleson published a study of the effects that circulation times have on the uptake of an inhaled anaesthetic. He found that the conventional approach causes systematic errors in the computed uptake of low-solubility agents, in the arterial concentration of high solubility agents, and in the tissue concentrations of all other agents. However these errors were only found to be important in the first minute or so after a change in the inhaled anaesthetic concentration has been made (Mapleson 1973).

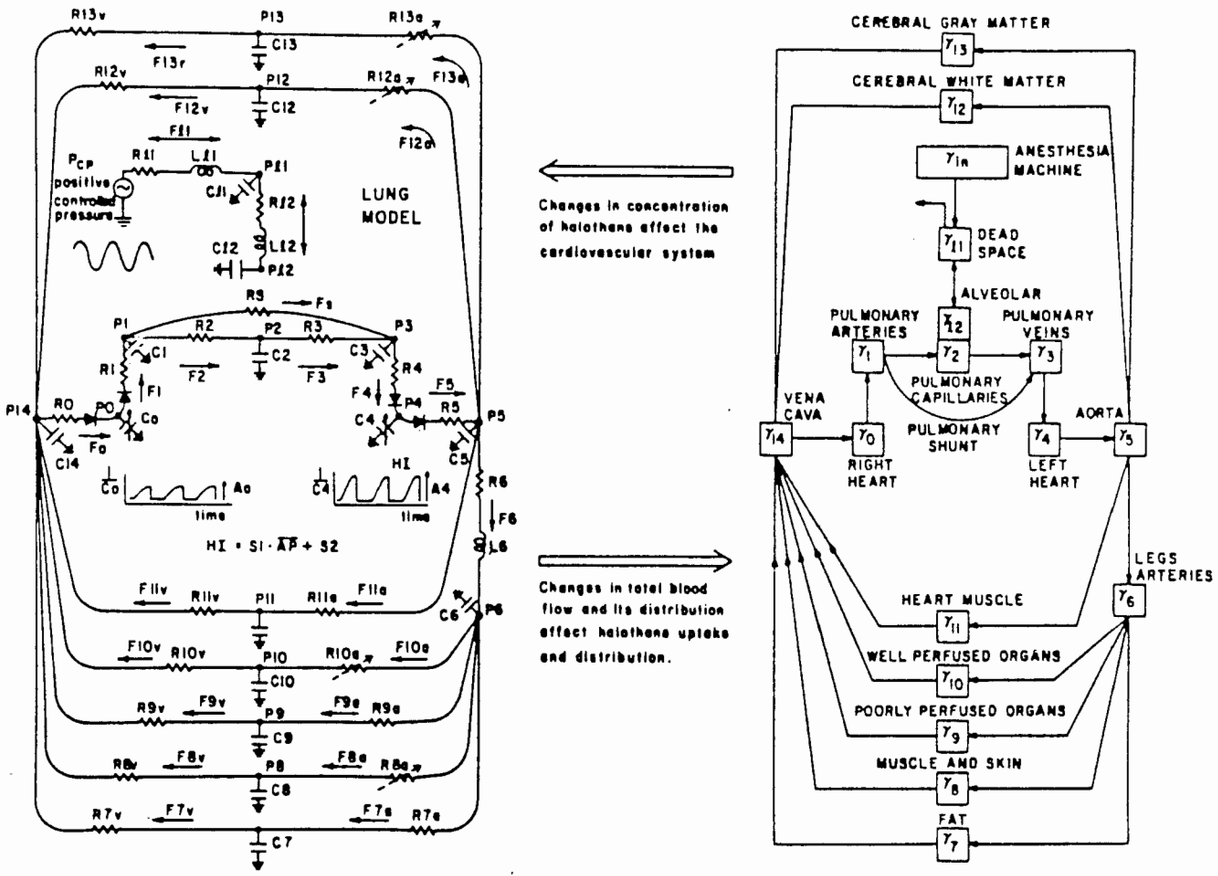


Figure 1.19 An 18 compartmental non-linear multiple model. The diagram shows a generalised scheme of the basic model. *Right*, the 18 compartmental model for the uptake and distribution of halothane. *Left*, the pulmonary (two segments) and cardiovascular (15 segments) models in electrical analogue form. The open arrows between the sub-models indicates the interactions. The halothane concentration in one compartment or a combination of three compartments (arterial blood, cerebral grey matter, and/or heart muscle) affects several cardiovascular parameters: 1) the slope (S1) and set value (S2) of the baroreceptor response, 2) amplitudes (A4 and A0) for the reciprocal compliances of left and right ventricles, and 3) regional vascular resistances, as indicated by the dashed arrows. The solid arrows in the lung/cardiovascular model represent the flow of air or blood, the latter out of the left heart into the aorta, thence into the regional arteries, the veins, the venae carvae, and right heart. R_s and F_s represent the left-to-right shunt, which is adjustable or controllable. γ_n = halothane concentration in the nth compartment in volumes percent; P_n = blood or air pressure in nth segment; F_n = blood or air flow in the nth segment; R_n = viscous flow resistance in nth segment (resistors); L_n = fluid inertance for nth segment (inductors); and, C_n = wall compliance for nth segment (capacitors). (Reproduced from Fukiu and Smith 1981a)

The most complete model that has been developed to date is a hybrid computer multiple model that consists of an 18 compartmental model of the uptake and transport of halothane, a 2 compartmental breath-by-breath pulmonary model, and a 15 compartmental beat-to-beat cardiovascular model that includes a baroreceptor heart rate loop (Fukiu and Smith 1981a). Figure 1.19 shows a schematic diagram of this model. The transport model was simulated on a digital mainframe computer and the pulmonary and cardiovascular models were simulated on an analogue computer that was connected to the digital computer by means of analogue to digital (A/D) converters. In a second paper an 18 compartmental model of the mass transport of carbon dioxide was included to give a model that can simulate the effect of carbon dioxide and halothane on the ventilation rate, heart rate, blood pressure, cardiac output and regional vascular resistances. A schematic diagram of the model is shown in figure 1.20 (Fukiu and Smith 1981b).

In 1982 a 3 compartmental linear model based on the solubility of anaesthetic gas in water and oil was developed to predict the alveolar to inspired concentration ratio (Tanner 1982). The differential equations of the model were solved exactly and a solution of the form

$$Fa/Fi = F/D + Ae^{r1t} + Be^{r2t} + Ce^{r3t}$$

resulted. This equation is easily solved on a hand held programmable calculator and the constants are supplied for 6 different agents.

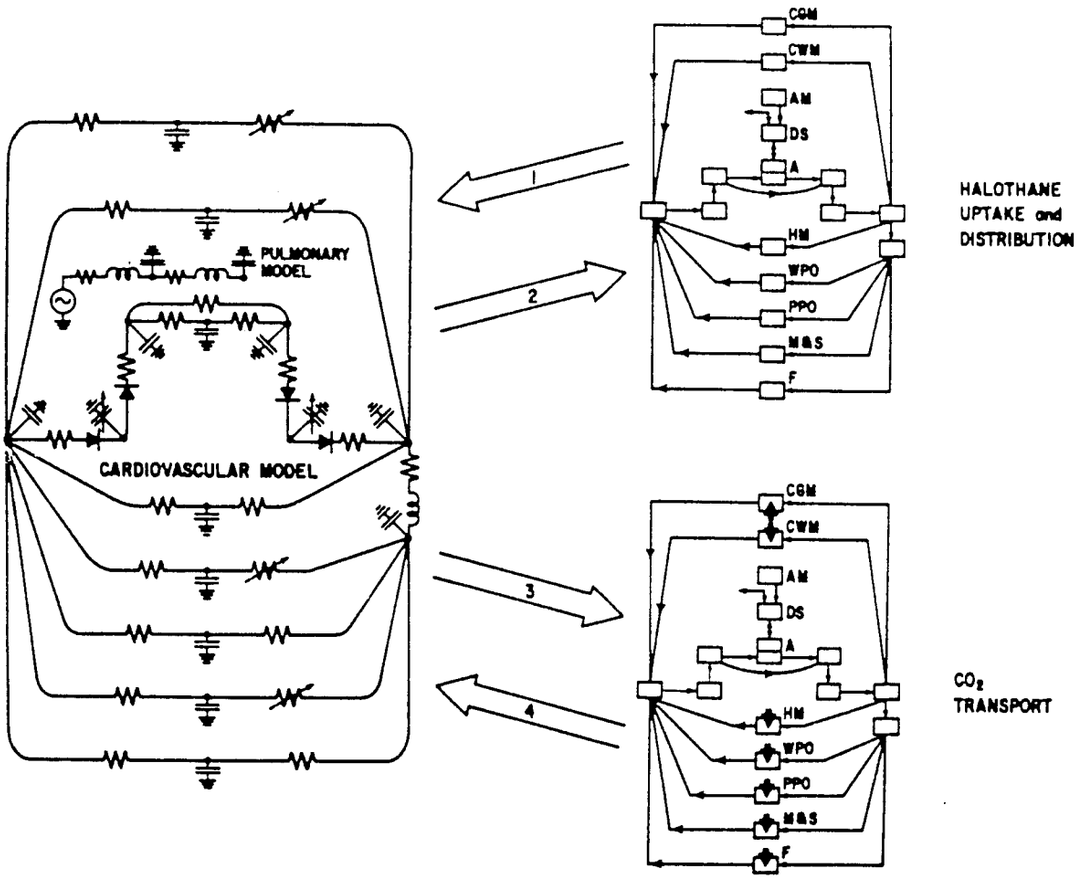
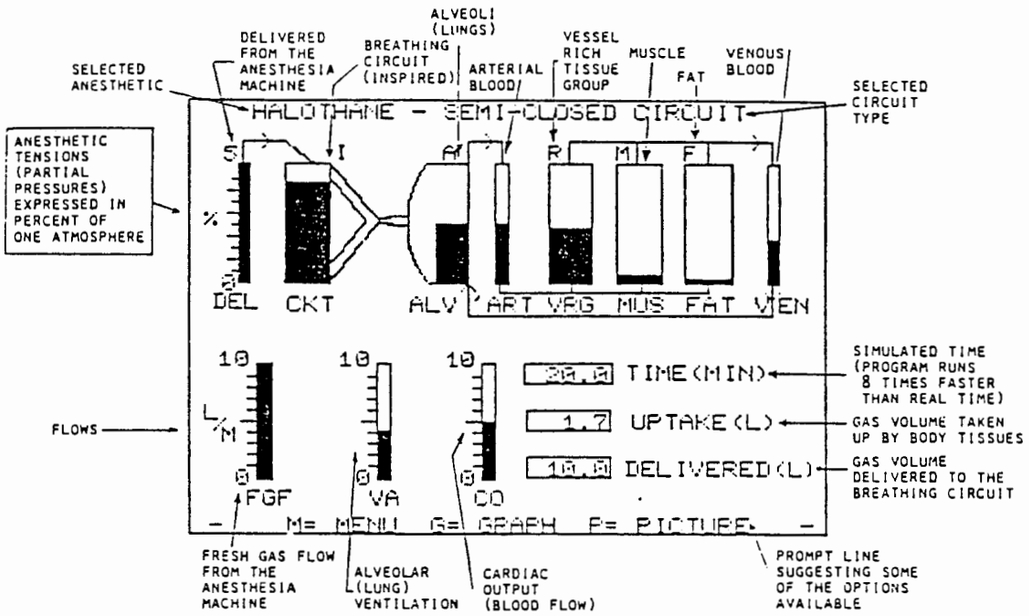
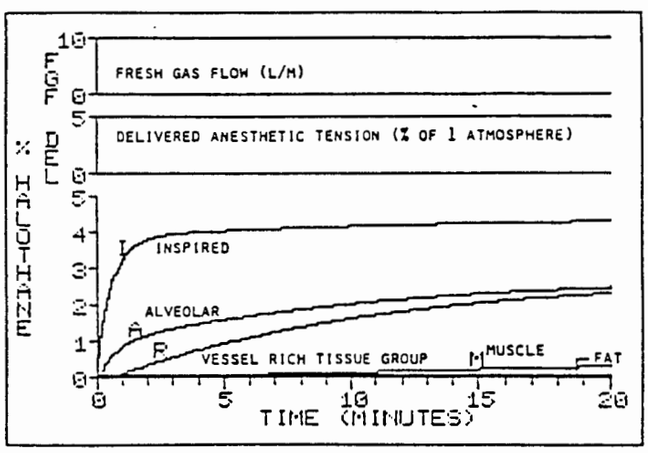


Figure 1.20 The 18 compartment model with the carbondioxide transport model included. *Arrow 1* : changes in halothane concentration affect the cardiovascular and respiratory systems. *Arrow 2* : changes in ventilation, as well as total blood flow and its distribution, affect halothane uptake and distribution. *Arrow 3* : Changes in ventilation, as well total blood flow and its distribution, affect CO_2 transport. *Arrow 4* : Changes in P_{CO2} affect the cardiovascular and respiratory systems. CGM = cerebral grey matter; CWM = cerebral white matter; AM = anaesthesia machine; DS = dead space; A = alveolar; HM = heart muscle; WPO = well perfused organs; PPO = poorly perfused organs; M&S = muscle and skin; and, F = fat. The CO_2 model is similar to the halothane mass-transport model. (Reproduced from Fukiu and Smith 1981b).

A 7 compartmental non-linear multiple model incorporating the effect of halothane on stroke volume, heart rate, blood pressure, peripheral blood flow, and ventilation was developed for use on a mainframe digital computer to teach undergraduates the pharmacokinetics of the uptake and distribution of halothane (Heffernan et al 1982a). The program can be used to produce rough graphical output on an



Completed pictorial display after a 20 minute halothane anesthetic.



Completed graphical display after administering a 20 minute anesthetic of 5% halothane with fresh gas flow 10 l/m. From top to bottom: 'FGF' fresh gas flow, 'DEL' delivered tension from the anesthesia machine. Below, anesthetic tension, expressed in percent of one atmosphere, in various locations: 'I' inspired gas, 'A' alveolar gas, 'R' vessel Rich group tissue, 'M' muscle tissue, and 'F' fat tissue.

Figure 1.21 "Gasman" simulator display screens. The operator can view changing parameters in bargraph form or as a linegraph to obseve trends. (Reproduced from Philip 1984).

alphanumeric terminal. Interaction with the model is available via the keyboard and the student may change the simulation speed, inspired halothane concentration, and fresh gas flow rate. Up to 9 model parameters can be graphed at a time. Four forms of the program are available : one with no changes in respiration or circulation, one with cardiovascular effects of halothane included, one with respiratory effects only and one with both of these effects combined. The student can study the importance of the effect of halothane on the circulatory and respiratory systems by comparing results from the different models. An evaluation of the effectiveness of the program for teaching the uptake and distribution of halothane showed a significant improvement in students understanding of the process (Heffernan et al 1982b).

"Gasman" a model based on Egers model features advanced pictorial and graphical simulation of anaesthetic uptake and distribution (Philip 1984). The student can choose one of 4 anaesthetic agents, 2 breathing circuits, and can set the alveolar ventilation rate and cardiac output. Figure 1.21 illustrates the graphics screens generated by the program.

Thompson et al (1986) went a few steps further. They designed a gas uptake simulator that includes a closed circle breathing circuit. The effects of physiological conditions, delivery parameters, and delivery methodologies can be evaluated. Interaction with the model is via the TV screen and a mouse which is used to move a pointer around on the screen. Two independent patient simulations can be executed at the same time allowing differences to be quickly seen. A

series of 12 computer assisted instruction lessons come with the simulator. The model is based on the square root of time model described by Lowe and Ernst (1981).

SUMMARY OF LITERATURE REVIEW.

The literature review indicated that the majority of incidents in anaesthesia are caused by human error, mechanical malfunctions being a fairly rare occurance. Human error may be due to negligence, or due to a lack of experience in coping with a certain situation. A lack of experience indicates a deficiency in training.

The review showed that only one full scale simulator has been developed for anaesthetic training, while a number of simulators have been developed for training in cardiopulmonary resuscitation and intubation techniques. Thus there is scope for developing a simulator for anaesthetics training.

Any simulator consists of a model of the system being simulated. The literature review showed that researchers have been developing models of the uptake and distribution of inert and anaesthetic agents for a number of years. Most of these models were used to try and understand the pharmacokinetics of the uptake and distribution of these gases. Some of the more recent models have been used in computer programs for teaching these principles to students by allowing them to adjust input parameters and observe the results as a graph on a screen. Table 1.2 summarises the models that were described in the review.

Author	Year	Type of model	Comp's	Comment
Kety	1951	Linear	2_a	First to include gas exchange at the lung.
Mapleson	1963	Linear	9 and 4	Tested various tissue compartment combinations.
Eger	1963	Linear	6	First to run model on a digital mainframe computer.
Beneken	1968	Lumped circuit	-	First to define the concept of a multiple model.
Ashman	1970	Non-linear	6	First to include the effect of anaesthetic agent on the cardiovascular system.
Zwart	1970	Non-linear	12	Includes the effect of agent on regional blood flow.
Munson et al	1972	Non-linear	6	Includes the effect of anaesthetic on ventilation and cardiac output.
Cowles et al	1973	Linear	5	A simplified method of running a model on a computer is described.
Mapleson	1973	Linear	6	3 models tested with blood circulation times taken into consideration.
Fukiu +Smith	1981	Non-linear multiple model	18	A combination of digital and analogue computers were used to run the model. A CO2 sub-model was also included.
Tanner	1982	Linear	3	A simple model for use on a hand held programmable calculator.
Heffernan	1982	Non-linear multiple	10	The first model incorporated into a software program for teaching uptake and distribution concepts. The software is interactive and produces rough graphics. The program runs on a mainframe computer.
Philip	1984	Linear	6	A teaching program for the IBM PC. High resolution graphics, and a selection of anaesthetic agents available.
Thompson	1986	Linear	11	A teaching program for the IBM PC. Simulates the use of a closed breathing circuit. Advanced graphics and supports the use of a mouse. A series of tutorials are also available.

[a] The number of compartments includes the lung compartment.

Table 1.3 A summary of models of the uptake and distribution of anaesthetic gases.

1.4 OBJECTIVES OF STUDY.

It is felt that there is some scope for improving anaesthetic training by developing a computer program that is capable of simulating the uptake and distribution of the anaesthetic agent halothane. The literature survey showed that a number of such programs have already been developed. One of these programs is based on an advanced non-linear multiple model but has very rough graphical output and is meant to run on a mainframe computer (Heffernan et al 1982a). The other two programs run on personal computers but are based on linear models that neglect the effect of the anaesthetic agent on the cardiovascular system (Philip 1984 and Thompson et al 1986). It is felt that an improved program for teaching the uptake and distribution of halothane can be developed. A generalised non-linear compartmental multiple model of the uptake and distribution of halothane can be developed by combining features of a number of existing models. The program can be written so as to allow any parameter of the model to be viewed or graphed during a simulation. This would allow the model to be tested, evaluated, and updated at a future stage. At the same time the program could be used for teaching purposes.

The primary objectives of this thesis are :

1) To review the literature covering anaesthetic training aids, including simulators and models of the uptake and distribution of anaesthetic agents.

2) To describe a generalised physiological model of the uptake and distribution of halothane that will be suitable for implementation on a personal computer.

3) To develop software that will allow the model to be tested, evaluated and updated if neccesary.

4) To develop software that can be used to teach the pharmacokinetics of the uptake and distribution of halothane to undegraduates, by allowing them to control the flow and concentration of anaesthetic gas to the model and observe the results in graphical form on the computer screen.

5) To make recomendations for further studies toward improving the simulator.

1.5 PRESENTATIONS.

COOPER RA, MORRELL DE, AMOORE JN.
Preliminary development of an anaesthetic training simulator.
Poster, Annual congress, Association of University Anaesthetists, Cape Town, March 1986.

CHAPTER 2

THE DEVELOPMENT OF A GENERALISED MULTIPLE MODEL

CHAPTER 2

THE DEVELOPMENT OF A GENERALISED MULTIPLE MODEL.

When an anaesthetic gas is introduced into the inspired air of a patient the tissues of the body take a finite time to equilibrate with the inspired concentration. A number of physical and physiological processes influence the rate at which this equilibrium is reached. This chapter reviews the factors which affect the uptake and distribution of an anaesthetic gas. A generalised n compartmental non-linear multiple model is then described following the approach used by Beneken (1972) for the uptake and distribution sub-model, by Zwart et al (1972) for the cardiovascular sub-model, and by Heffernan et al (1982) for the circle breathing circuit. The resulting differential equations can then be used for simulation purposes.

2.1 FACTORS AFFECTING THE UPTAKE AND DISTRIBUTION OF ANAESTHETIC AGENTS.

The route followed by an anaesthetic gas from the inspired air to the tissues is as follows. The gas is inspired and mixes with the air in the lungs. The diluted gas then diffuses through the alveolar membranes to equilibrate with the pulmonary blood and thus enters into the blood circulation. In the systemic circulation it is distributed via the peripheral arteries to the individual tissues of the body. At the tissue level the gas diffuses across the capillary membrane, interstitial fluid, and cellular membranes of the various body tissues. Some fraction of the original gas concentration of the arterial blood is then

returned via the venous blood to the lungs. As the various body tissues become saturated with the anaesthetic agent, less anaesthetic is removed from the blood by the tissues and the mixed venous anaesthetic blood concentration rises. This lowers the concentration gradient across the alveolar membrane which slows down the rate at which the anaesthetic agent is absorbed. This in turn allows ventilation to drive the alveolar concentration of the gas a little higher.

Figure 2.1 shows a typical trace of the rate of change of the alveolar concentration of halothane (as a fraction of the inspired concentration) versus time with constant ventilation. The rate at which the alveolar concentration approaches the inspired concentration (and thus the shape of the curve in figure 2.1) is governed by the rate at which the gas is delivered to the alveoli and by the rate at which it is removed from them.

Delivery to the lungs is determined by the following factors:

- The inspired concentration of anaesthetic.
- The alveolar minute volume.
- The functional residual capacity of the lungs.

A high inspired concentration allows the alveolar concentration to approach the inspired concentration more rapidly. Eger (1963b) called this the concentration effect. Figure 2.2 shows this effect for different concentrations of N_2O and ether. The other factor that governs the delivery of gas to the alveoli is

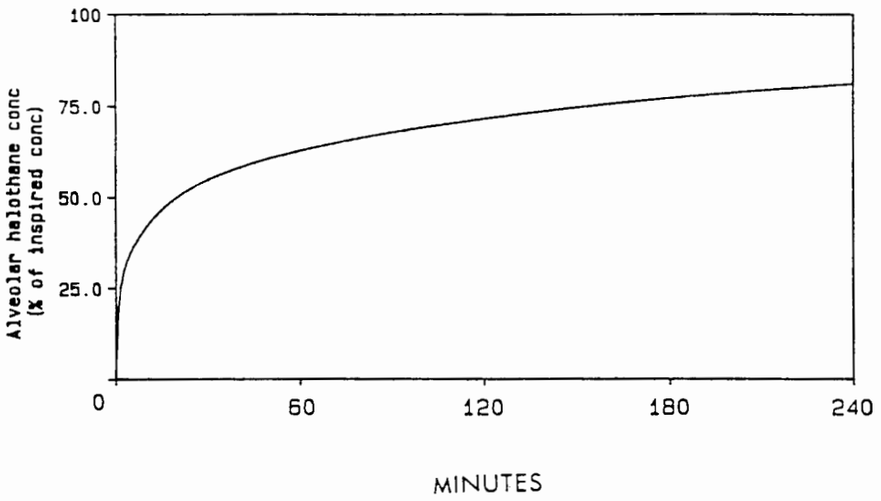


Figure 2.1 A typical trace of the rate of change of the alveolar concentration of halothane.

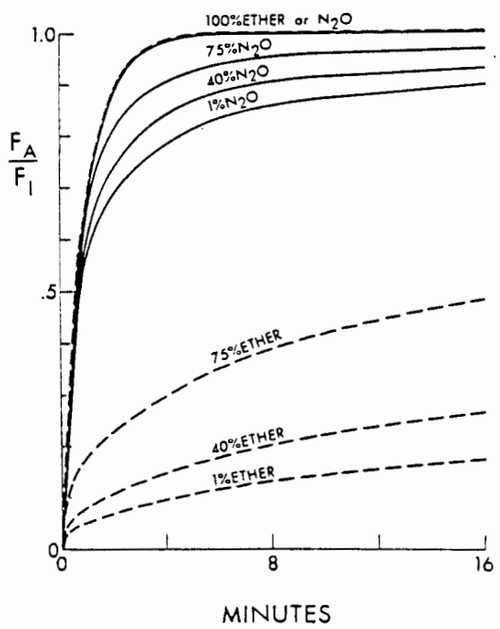


Figure 2.2 The concentration effect. Increasing the inspired concentration (indicated by the percent figures) from 1 to 40 to 75 to 100 percent, increases the rate of rise of the alveolar concentration of both ether and nitrous oxide. At 100 percent inspired concentration, the rate of rise is identical for both agents and equals the wash-in rate of the lungs. (Reproduced from Eger 1974).

the alveolar minute volume. The greater the ventilation rate the more rapid the approach of the alveolar concentration to the inspired concentration.

This can be seen in figure 2.3 for ventilation rates of 2, 4, and 8 litres per minute with a constant inspired concentration of halothane. However, the effect of alveolar minute volume is limited by the volume of the functional residual capacity (FRC) of the lungs. The larger the FRC the slower the wash-in of a new gas due to dilution.

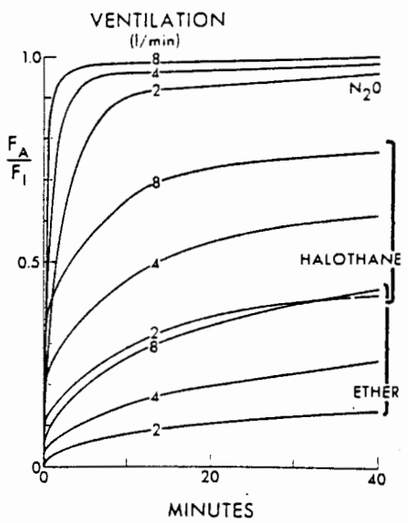


Figure 2.3 The effect of alveolar ventilation on uptake. The alveolar rate of rise toward the inspired concentration (Fa/Fi) is accelerated by an increase in alveolar ventilation from 2 to 4 and from 4 to 8 litres/minute (constant cardiac output). The ratio of lowest to highest anaesthetic dose produced by these differences in ventilation is greatest with the most soluble anaesthetic, ether; less with the anaesthetic of intermediate solubility, halothane; and least with the least soluble anaesthetic, nitrous oxide. (Reproduced from Eger 1974).

The gain in alveolar concentration due to ventilation is opposed by a minor loss of anaesthetic to the lung tissue

and a major loss to the blood passing through the lungs. The uptake of anaesthetic gas from the alveoli is determined by the following factors :

- Solubility of anaesthetic in blood .
- Blood flow through the lungs.
- Alveolar to mixed venous blood partial pressure difference.

Diffusion of anaesthetic gas through the walls of the alveoli into the pulmonary blood (uptake) is determined by the solubility of the gas in blood, and by the difference in partial pressure (concentration) of the gas between the alveoli and pulmonary blood.

The solubility coefficient S of a gas in a liquid can be expressed as the ratio of the concentration C of the gas dissolved in the liquid to the partial pressure P of the gas in the gas phase with which the solution is in equilibrium.

$$S = C/P \qquad (2.1)$$

The most commonly used definition of S in anaesthesia is the Ostwald solubilty coefficient λ defined as the ratio of the volume of gas absorbed to the volume of the absorbing liquid, at any specified temperature and pressure. For liquids and tissues in the body, C in equation 2.1 is given in volumes of gas at 37 deg C, and 1 atmosphere pressure per volumes of liquid, and P is given in standard atmospheres.

A liquid/gas partition coefficient is defined as the ratio

between the concentration of a gas in the liquid and the concentration of the gas in the gas phase with which it is in equilibrium. If the concentrations are both expressed in units of content per unit volume, then the liquid/gas partition coefficient is numerically equal to the ostwald solubility coefficient and is unitless. Partition coefficients also exist between two liquids : ie the tissue/blood coefficient is the ratio between the concentration in the tissue and in the blood with which it is in equilibrium (Steward et al 1973).

The larger the liquid/gas solubility coefficient or partition coefficient, the more gas a particular liquid can absorb at a particular concentration. Thus an anaesthetic gas such as ether which has a large blood/gas solubility coefficient will diffuse into the pulmonary blood more readily thus increasing the uptake of the gas. This in effect slows down the rate at which the alveolar concentration approaches the inspired concentration. This is shown in figure 2.4 for four anaesthetic agents with different blood solubilities. Note that the concentration of the agents in the alveoli is given as a fraction of the inspired concentration.

Cardiac output is also an important factor in governing the uptake of gas from the lungs. The greater the cardiac output, the more rapidly the absorbed gas is removed from the lungs to the tissues. This is an important factor as most anaesthetic gases have a depressant effect on the contractility of the heart (Stoelting 1984, Eger et al 1968, Eger et al 1970). This causes a decrease in cardiac output which affects the uptake of the anaesthetic agent. This

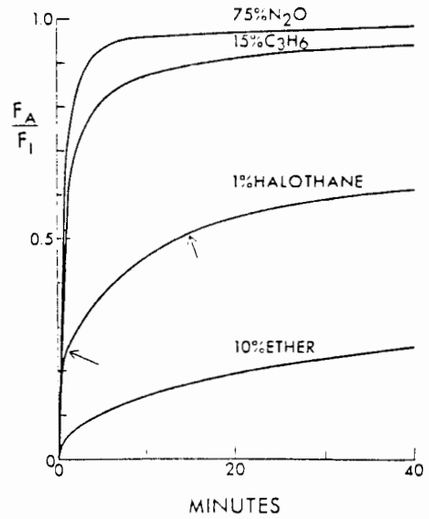


Figure 2.4 The effect of solubility on uptake. The approach of the alveolar (F_A) concentration to the inspired (F_I) varies inversely with solubility. It is slow with a very soluble agent such as ether but quickly nears 100% (1.0) with nitrous oxide or cyclopropane. Despite differences in their heights, the curves for ether, halothane and cyclopropane follow the same three part pattern: a rapid upswing (to the first arrow in the case of halothane), a slower continuing upsweep (to the second arrow), and a continuing slower rise thereafter. (Reproduced from Eger 1974).

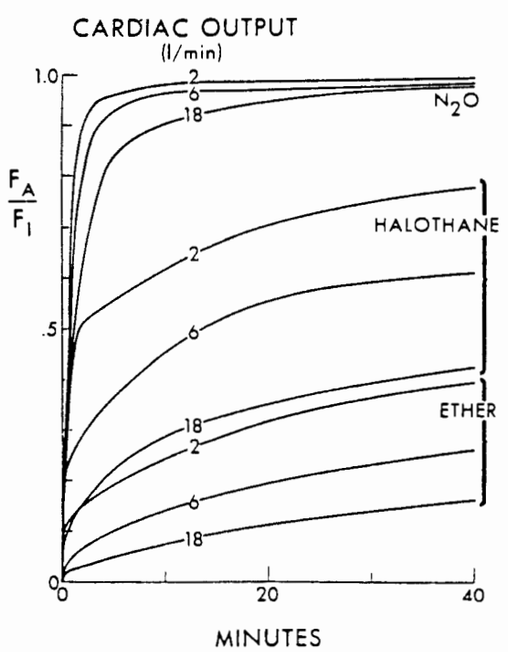


Figure 2.5 The effect of cardiac output on uptake. If ventilation is held constant, then increases in cardiac output of from 2 to 6 to 18 litres/minute will increase uptake and consequently retard the rate of rise of the alveolar concentration (F_A) toward the concentration inspired (F_I). These graphs assume that changes in cardiac output proportionately affect all tissues. (Reproduced from Eger 1974)

effect is shown in figure 2.5 for halothane, ether, and N_2O with cardiac outputs of 2, 6, and 18 litres/minute. Inspired concentration and ventilation rate are both held constant.

As the concentration of anaesthetic gas in the tissues starts to rise, the concentration of anaesthetic agent in the tissue venous blood also starts to rise. Venous blood is returned to the lungs via the pulmonary arteries. A rise in the concentration of anaesthetic agent in the venous blood lowers the concentration gradient between the alveoli and the mixed venous blood of the lungs. This has the effect of decreasing the diffusion rate which decreases uptake. A decreased uptake allows ventilation to drive the alveolar concentration a little higher.

Uptake of anaesthetic by the lungs is equal to the sum of the uptake of the agent by the various body tissues. The uptake by the tissues is governed by the following factors:

- Solubility of anaesthetic in tissue.
- Tissue blood flow.
- Arterial to tissue partial pressure difference.

The amount of anaesthetic gas that a tissue can absorb is expressed as the capacity of that tissue. Tissue capacity for an anaesthetic gas is defined to be the product of the tissue partition coefficient and tissue volume.

$$\text{Tissue capacity} = \lambda_{tg} \cdot V_t \quad \text{(litres)} \qquad (2.2)$$

λ_{tg} = Tissue/gas partition coefficient.

V_t = Volume of tissue (litres).

The relationship between the tissue capacity and the tissue blood flow determines the rate at which the tissue can absorb an anaesthetic gas. Tissue blood flow is an important factor to consider. A tissue with high blood flow but a small capacity will quickly become saturated. However a tissue with a small blood flow and the same capacity will take much longer to reach saturation. Anaesthetic gases can cause vasodilation or vaso-constriction in certain tissues thus changing the rate at which the blood can deliver the anaesthetic agent to them. As each tissue in the body becomes saturated it removes less gas from the arterial blood supplying it, and the mixed venous partial pressure gradually rises thus decreasing uptake from the lungs.

Eger (1974) found it convenient to consider four tissue groups for the purpose of explaining uptake by the tissues. These are the vessel rich group (brain, heart, kidneys, splanchnic bed, and endocrine glands), the muscle group, the fat group, and the vessel poor group (bony, cartilagenous, and ligamentous tissues). Figure 2.6 shows the simulated uptake of halothane and N_2O by these tissue groups.

Note that uptake by the vessel rich group (VRG) predominates

during the first few minutes because of its high blood flow. However it soon becomes saturated, which leaves the muscle group (MG) and the fat group (FG) as the sole determinants of uptake. Uptake by the vessel poor group is neglected as it is very small. The total uptake by the lungs continuously decreases as the various tissues approach saturation.

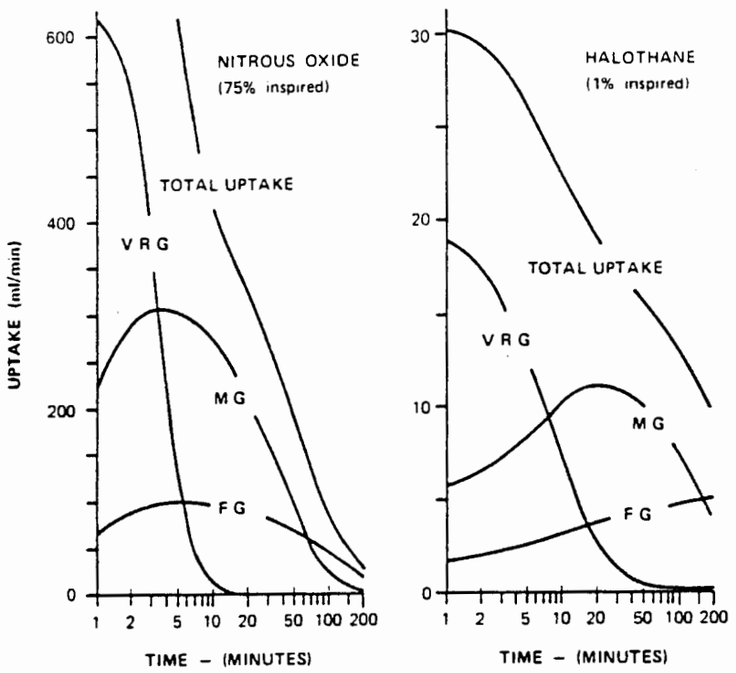


Figure 2.6 The rate of uptake by different compartments. Total uptake is the sum of uptake by individual body tissues. The uppermost curve for each anaesthetic is the sum of the three curves beneath it (with a slight time lag). The curves for nitrous oxide and halothane are identical in shape to those for all anaesthetics. Uptake progressively decreases with duration of anaesthesia and with saturation of the tissues. The order of saturation is always the vessel rich group (VRG) first then the muscle group (MG), and the fat group (FG) last. (Reproduced from Eger 1974).

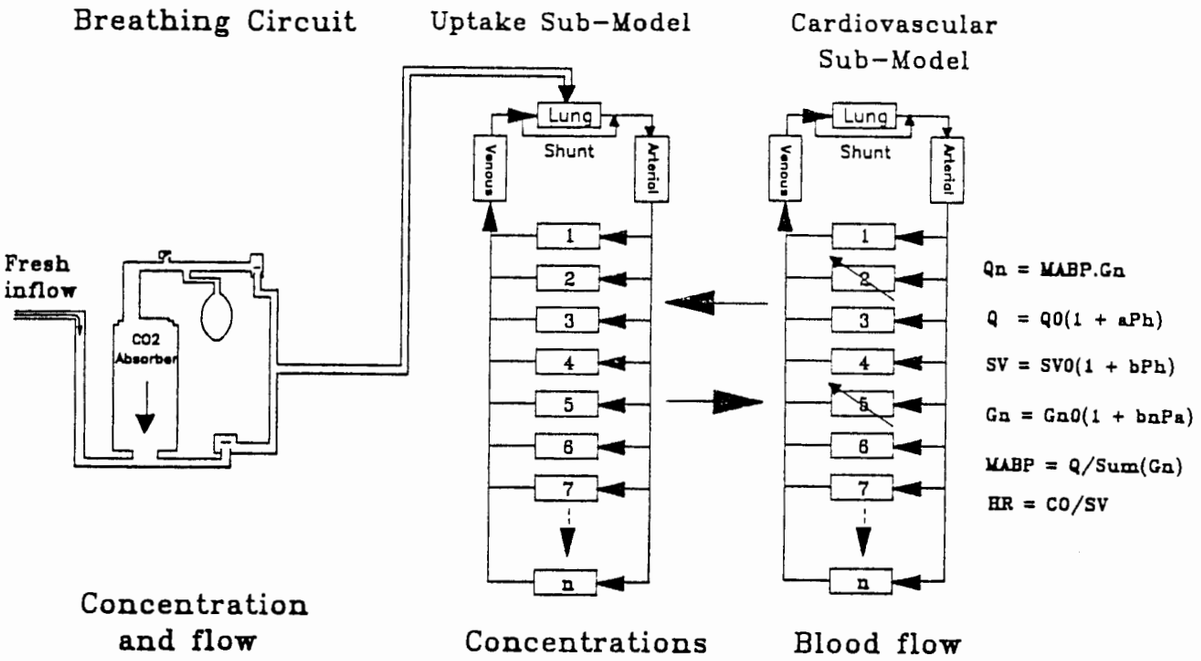
2.2 MODELLING AND COMPUTER SIMULATION.

For the purposes of this thesis modelling can be defined as the construction of a mathematical representation of a real system. This model would be an approximation of this system and could be used to simulate the response of the system to various external disturbances. For the purpose of an anaesthetic training simulator a mathematical model of the human cardiovascular and respiratory systems is required which will realistically respond to a simulated anaesthetic procedure.

In this section a generalised n compartmental multiple model of the uptake and distribution of halothane, based on a model developed by Zwart et al (1972), is described. Figure 2.7 shows a schematic diagram of the model which consists of 3 seperate sub-models, namely the breathing circuit, the uptake sub-model and the cardiovascular sub-model. In the following sections a mathematical description of each sub-model will be presented, which will result in a number of differential equations that can be used to simulate the response of these systems to halothane.

2.3 THE UPTAKE AND DISTRIBUTION SUB-MODEL.

The uptake and distribution sub-model is concerned with the amount of anaesthetic agent entering and leaving each of the body tissue compartments, and with the concentration of anaesthetic agent in these compartments at any time. Blood flow through each compartment is controlled by the cardiovascular sub-model, which in turn is affected by



Pn	= nth compartment halothane partial pressure	(Frac atm)
Qn	= Blood flow through the nth compartment	(l/min)
Gn	= Conductance of the nth compartment	(l/min/mmHg)
MABP	= Mean arterial blood pressure	(mmHg)
HR	= Heart rate	(Beats/min)
SV	= Stroke volume	(l)
SV0	= stroke volume in the unanaesthetised state	(l)
Q	= Cardiac output	(l/min)
Q0	= Cardiac output in the unanaesthetised state	(l/min)
Ph	= Heart muscle halothane partial pressure	(Frac atm)
Pa	= Arterial halothane partial pressure	(Frac atm)
a,b,bn	= Linear relationship constants	

Figure 2.7 A generalised n compartment multiple model. The cardiovascular model is concerned with blood flow to the various compartments, the uptake model is concerned with the concentration of anaesthetic in the various compartments. The large arrows between the two models indicate that they influence one another. All other arrows indicate the direction of gas and blood flows. The semi-closed breathing circuit model determines the concentration of anaesthetic gas being inspired.

halothane concentrations in the uptake sub-model.

The following simplifying assumptions are made :

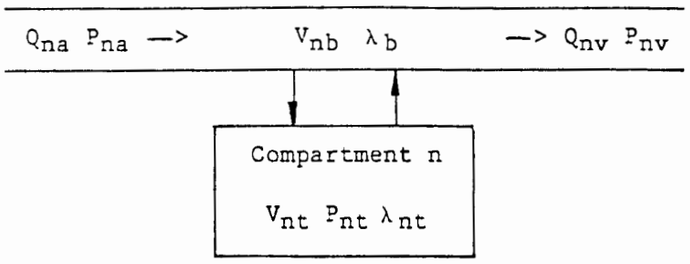
1/ There is no metabolism or cutaneous loss of anaesthetic agent. (In reality there may be up to 25% metabolism).

2/ There is no exchange of anaesthetic agent between compartments except by transport in the circulatory system.

3/ Compartments are assumed to be perfect mixing chambers. ie. equilibration within the compartment is instantaneous.

The body consists of many different tissue types, each with different blood flows, gas solubilities and capacities. For the purposes of modelling the uptake and distribution of anaesthetic agents it is convenient to group tissues with similar blood flows, anaesthetic gas tissue solubilities and capacities into what is termed a body compartment (Kety 1950, Eger 1974, Beneken 1972). A general body compartment "n" is shown in figure 2.8.

Arterial blood flows into this compartment at some flow rate Q_{na}, and leaves the compartment as venous blood at some flow rate Q_{nv}. The blood inside the compartment has a volume V_{nb} and has a gas concentration P_{nb} with a blood-gas partition coefficient λ_b. The tissue in the compartment has a volume V_{nt} and has a gas concentration P_{nt} with a tissue-gas partition coefficient λ_{nt}.



Q_{na} - Arterial blood flow into compartment n (l/min)
Q_{nv} - Venous blood flow out of compartment n (l/min)
P_{na} - Partial pressure of anaesthetic in arterial blood (Frac Atm)
P_{nv} - Partial pressure of anaesthetic in venous blood (Frac Atm)
P_{nt} - Partial pressure of anaesthetic in tissues of compartment n (Frac Atm)
V_{nb} - Volume of blood in compartment n (litres)
V_{nt} - Volume of tissue in compartment n (litres)
λ_b - Blood/gas partition coefficient of anaesthetic
λ_{nt} - Tissue/gas partition coefficient of anaesthetic

Figure 2.8 Schematic diagram of a compartment which consists of both a volume of tissue and a volume of blood.

2.3.1 THE CONCENTRATION OF ANAESTHETIC IN A COMPARTMENT.

If an amount α_n litres of an anaesthetic gas A is dissolved in a compartment n which has a total capacity V_{ntot} litres for A (Equation 2.2), then the concentration (or partial pressure) P_n of A in the compartment is given by:

$$P_n = \frac{\alpha_n}{V_{ntot}} \quad \text{(Fraction Atm)} \qquad (2.3)$$

Or

$$\alpha_n = P_n . V_{ntot} \quad \text{(litres)} \qquad (2.4)$$

A compartment consists of both a volume of tissue and a volume of venous blood, each of which has some finite capacity for the anaesthetic gas. Thus equation 2.4 can be modified to calculate the amount of anaesthetic in the compartment n as follows :

$$\alpha_n = P_{nb} \cdot V_{nbtot} + P_{nt} \cdot V_{nttot} \qquad (2.5)$$

(Blood) + (Tissue)

where V_{nbtot} is the capacity of the blood in the compartment for the anaesthetic. Substituting equation 2.2 for tissue capacity into equation 2.5 yields :

$$\alpha_n = P_{nb} \cdot \lambda_b \cdot V_{nb} + P_{nt} \cdot \lambda_{nt} \cdot V_{nt} \qquad (2.6)$$

Assuming that equilibration of anaesthetic partial pressure between the blood and tissue in a compartment is very rapid (Kety 1951, Mapleson 1963, Beneken 1972) then $P_{nb} = P_{nt} = P_n$. Thus by rearranging equation 2.6 the following equation for the concentration of an anaesthetic in the compartment is obtained:

$$P_n = \frac{\alpha_n}{\lambda_b \cdot V_{nb} + \lambda_{nt} \cdot V_{nt}} \quad \text{(Frac Atm)} \qquad (2.7)$$

2.3.2 THE AMOUNT OF ANAESTHETIC IN A COMPARTMENT.

Assuming no direct diffusion of anaesthetic between the various compartments of the body (Mapleson 1963, Beneken 1972), then the only way that the anaesthetic gas can arrive at or leave the compartment is via the blood that flows into and out of the compartment.

The amount of anaesthetic agent that flows into the compartment over a certain period of time t is given by the following equation (Beneken 1972) :

$$\alpha_n(t) = \alpha_n(0) + \int_0^t (P_{na} \cdot \lambda_b \cdot Q_{na} - P_{nv} \cdot \lambda_b \cdot Q_{nv})dt \qquad (2.8)$$

(Arterial) - (Venous)

$\alpha_{n(0)}$ is the amount of anaesthetic that was already present in the compartment at time t = 0. Under the conditions of non-pulsatile blood flow (blood circulation times are assumed to be much shorter than equilibration times), and a constant blood volume in the compartment, the blood flow into the compartment equals the blood flow leaving it and thus $Q_{na} = Q_{nv} = Q_n$. As equilibration between tissues and venous blood is assumed to occur very rapidly $P_{nv} = P_{nt} = P_n$. Equation 2.8 then simplifies to become :

$$\alpha_n(t) = \alpha_n(0) + \lambda_b \cdot Q_n \cdot \int_0^t (P_{na} - P_n)dt \qquad (2.9)$$

2.3.3 A GENERAL BODY COMPARTMENT.

Substituting the expression for α_n from equation 2.9 into equation 2.7 an equation for the concentration of an anaesthetic agent in a general compartment n as a function of time is obtained :

$$P_n(t) = P_n(0) + \frac{\lambda_b \cdot Q_n}{\lambda_b \cdot V_{nb} + \lambda_{nt} \cdot V_{nt}} \cdot \int_0^t (P_{na} - P_n)dt \qquad (2.10)$$

The time derivative of equation 2.10 yields a differential equation for the concentration of an anaesthetic gas in any general compartment n :

$$d(P_n)/dt = \frac{\lambda_b \cdot Q_n}{\lambda_b \cdot V_{nb} + \lambda_{nt} \cdot V_{nt}} \cdot [\, P_{na} - P_n \,] \qquad (2.11)$$

2.3.4 THE LUNG COMPARTMENT.

The lung compartment consists of both the lungs and the heart. Blood flows into the compartment via the venae cavae and out into the aorta. Note that the heart is not indicated on the diagram. Both the inspired air and the pulmonary blood flow carry gas to and from the lung compartment and thus equation 2.11 cannot be used directly to derive the anaesthetic gas concentration in this compartment. Figure 2.9 shows a typical lung compartment which has an alveolar ventilation rate of Q_{alv} litres/minute and a pulmonary blood flow of Q_l litres/minute. P_{ins} is the inspired anaesthetic concentration as a fraction of atmospheric pressure.

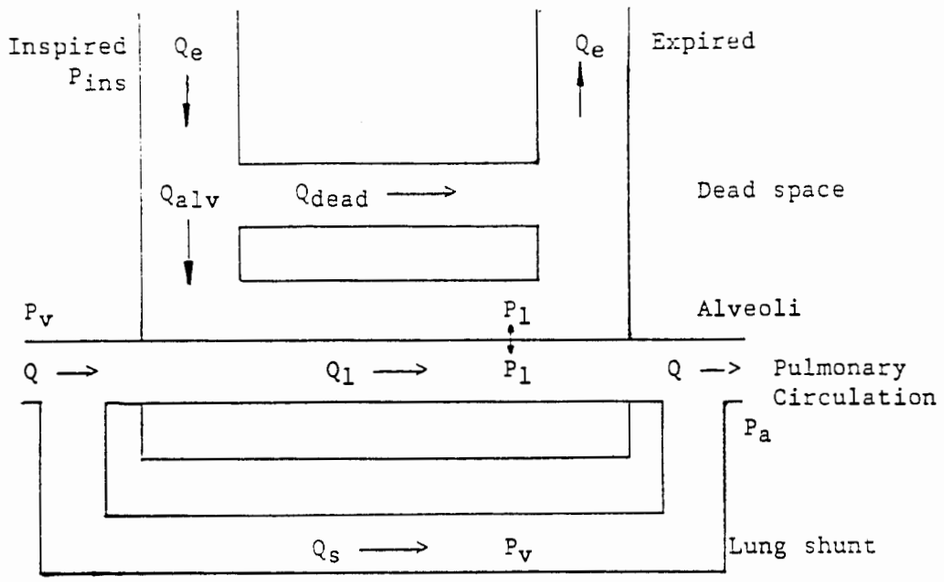
Expanding equation 2.9 to include the effects of both air and blood flows gives an equation for the amount of anaesthetic in the lung compartment at any time :

$$\alpha_l(t) = \alpha_l(0) + \int_0^t \underbrace{(P_{vl}.\lambda_b.Q_l - P_l.\lambda_b.Q_l)dt}_{\text{(Blood)}} + \underbrace{(P_{ins}.Q_{alv} - P_l.Q_{alv})dt}_{\text{(Air)}} \qquad (2.12)$$

This simplifies to :

$$\alpha_l(t) = \alpha_l(0) + \int_0^t [\lambda_b.Q_l \cdot (P_{vl} - P_l) + Q_{alv} \cdot (P_{ins} - P_l)]dt \qquad (2.13)$$

The capacity of the lung for an anaesthetic gas is a function of the lung tissue volume, pulmonary blood volume, and the volume of air in the lungs. Equation 2.5 must be modified to take the volume of air in the lungs into account:



P_{ins}	-	Inspired halothane partial pressure	(Frac atm)
P_v	-	Venous halothane partial pressure	(Frac atm)
P_l	-	Alveolar halothane partial pressure	(Frac atm)
P_a	-	Arterial halothane partial pressure	(Frac atm)
Q_e	-	Total minute ventilation rate	(l/min)
Q_d	-	Dead space minute ventilation rate	(l/min)
Q_{alv}	-	Alveolar minute ventilation rate	(l/min)
FRC	-	Functional residual capacity	(l)
V_{dead}	-	Dead space volume	(l)
V_{tidal}	-	Tidal volume	(l)
V_l	-	FRC + 1/2 V_{tidal}	(l)
Q	-	Cardiac output	(l/min)
Q_l	-	Blood flow rate through lung compartment	(l/min)
Q_s	-	Blood flow rate through the lung shunt	(l/min)

Figure 2.9 A schematic diagram of the lung compartment.

$$\alpha_l = P_l.\lambda_b.V_{lb} + P_l.\lambda_{lt}.V_{lt} + P_l.V_l \qquad (2.14)$$

where V_l = FRC + 1/2 Tidal volume (Zwart et al 1972). Note that the solubility of an anaesthetic gas in air is by definition equal to 1. Rearranging this equation yields the following equation for the concentration of an anaesthetic gas in the lung compartment :

$$P_l = \frac{\alpha_l}{\lambda_b \cdot V_{lb} + \lambda_{lt} \cdot V_{lt} + V_l} \qquad (2.15)$$

Substituting $\alpha_l(t)$ from equation 2.13 into equation 2.15 yields an equation for the concentration of anaesthetic gas in the lung at any time t :

$$P_l(t) = P_l(0) + \int_0^t \frac{[\lambda_b \cdot Q_l \cdot (P_{vl} - P_l) + Q_a \cdot (P_{ins} - P_l)]\, dt}{\lambda_b \cdot V_{lb} + \lambda_{lt} \cdot V_{lt} + V_l} \qquad (2.16)$$

The first derivative of equation 2.16 yields a differential equation for the alveolar anaesthetic concentration :

$$d(P_l)/dt = \frac{\lambda_b \cdot Q_l \cdot (P_{vl} - P_l) + Q_{alv} \cdot (P_{ins} - P_l)}{\lambda_b \cdot V_{lb} + \lambda_{lt} \cdot V_{lt} + V_l} \qquad (2.17)$$

2.3.5 THE ARTERIAL AND VENOUS COMPARTMENTS.

The arterial and venous blood compartments also have to be considered. The amount of anaesthetic in the venous compartment at any time t is given by :

$$\alpha_v(t) = \alpha_v(0) + \int_0^t (\lambda_b \cdot \sum_i Q_i \cdot P_i) - Q \cdot P_v)\, dt \qquad (2.18)$$

Where $\Sigma Q_i.P_i$ is the sum of all flows and concentrations from all compartments being drained by the venous system and Q is the total cardiac output in litres/minute.

By replacing α_n in equation 2.5 with equation 2.18 the following equation for the concentration of anaesthetic

in the venous blood is obtained :

$$P_v(t) = P_v(0) + \frac{1}{(\lambda_b.V_{vb} + \lambda_{vt}.V_{vt})} \int_0^t [\lambda_b.(\sum_i Q_i.P_i) - Q.P_v]dt \qquad (2.19)$$

The derivative of equation 2.19 yields the following differential equation for the concentration of anaesthetic in the venous blood:

$$d(P_v)/dt = \frac{\lambda_b}{\lambda_b.V_{vb} + \lambda_{vt}.V_{vt}} [(\sum_i Q_i.P_i) - Q.P_v] \qquad (2.20)$$

Similarly for the arterial compartment :

$$\alpha_a(t) = \alpha_a(0) + \int_0^t (\lambda_b.[Q_s.P_v + Q_l.P_l] - Q.P_a)dt \qquad (2.21)$$

(Shunt + lung) - (arterial)

Q_s is the blood flow past the lungs in a shunt as a percentage of cardiac output. The differential equation for the concentration of anaesthetic in the arterial blood is :

$$d(P_a)/dt = \frac{\lambda_b}{(\lambda_b.V_ab + \lambda_{at}.V_{at})} \left[Q_s.P_v + Q_l.P_l - \frac{Q.P_a}{\lambda_b} \right] \qquad (2.22)$$

As an approximation the arterial blood pool may be included in the lung compartment and the venous blood pool may be split up amongst the other tissue compartments in proportion to the fraction of the cardiac output that they receive (Mapleson 1973). The venous concentration of anaesthetic is then the sum of the concentrations from the various compartments multiplied by the fraction of cardiac output that flows through the various

compartments :

$$P_v = \frac{\sum_i P_i \cdot Q_i}{Q} \qquad (2.23)$$

Arterial concentration is equal to alveolar concentration, which is equal to the concentration of the lung tissue. If there is a lung shunt then this fraction that bypasses the alveoli must be taken into account :

$$P_a = \frac{Q_s \cdot P_v + Q_l \cdot P_l}{Q} \qquad (2.23)$$

2.4 THE NON-PULSATILE CARDIOVASCULAR SYSTEM SUB-MODEL.

The cardiovascular sub-model deals with the cardiac output and with the distribution of blood to the various compartments. The following assumptions are required to allow a simple model to be defined :

1/ Blood flow is assumed to be non-pulsatile as equilibration times are large compared to the cardiac cycle.
2/ Transport times are negligable for the same reason.
3/ There is no adaptation to the anaesthetic agent.
4/ No other factors other than the anaesthetic agent have any effect on the cardiovascular system.
5/ The anaesthetic agent in this case is halothane only.
6/ Baroreceptor action on heart rate is neglected.

Cardiac output and regional blood flow distribution vary in response to halothane administration. It is possible that this is either a local effect or mediated by the central nervous system (Beneken 1972).

The cardiac output is assumed to decrease linearly with increasing concentration of halothane in the brain grey matter compartment (Zwart et al 1972). The relationship is as follows :

$$Q = Q0(1 + a.P_g) \qquad (l/minute) \quad (2.24)$$

Q0 is the cardiac output in the unanaesthetised state and "a" is a constant that determines the slope of the relationship between cardiac output and the concentration of

halothane in the cerebral grey matter.

In a similar manner a relationship for the stroke volume SV can be defined :

$$SV = SV0(1 + b.P_g) \qquad \text{(litres)} \quad (2.25)$$

The values of "a" and "b" were calculated from data obtained by Eger et al (1970), and are given in APPENDIX A.

Heart rate may be calculated from the relationship :

$$HR = Q/SV \qquad \text{(Beats/minute)} \quad (2.26)$$

The blood flow through the nth compartment (Q_n) is given by the relationship :

$$Q_n = MABP \cdot G_n \qquad \text{(litres/min)} \quad (2.27)$$

where G_n, the conductance, is the inverse of the resistance of the compartment to the blood flow, and MABP is the mean arterial blood pressure. The conductance to blood flow of the nth compartment (G_n) is assumed to be a linear function of the arterial concentration of halothane (Zwart et al 1972) The relationship is as follows :

$$G_n = G_n0(1 + b_n \cdot P_a) \qquad \text{(l/min/mmHg)} \quad (2.28)$$

where G_n0 is the conductance of compartment n in the unanaesthetised state and b_n is a proportionality constant. The conductances of certain compartments such as the visceral, poorly perfused tissue, myocardial, fat and lung compartments are not affected by the concentration of halothane and b_n is set equal to zero in equation 2.28.

Mean arterial blood pressure (MABP) is equal to the cardiac output (Q) divided by the sum of the conductances of the n parallel compartments shown in figure 2.7:

$$MABP = \frac{Q}{\sum_{n} G_n} \qquad (mmHg) \qquad (2.29)$$

The stroke volume is proportional to pulse pressure (the difference between the diastolic and systolic pressures) only if the pressure-volume relations in the arterial system are constant and uniform among individuals, but linear from high to low pressure. This is not the case in reality due to changes in peripheral resistance and the compliance of the major arteries. However research by Hamilton and Remington in 1947 found a rough correlation between stroke volume and pulse pressure over a normal pressure range. This relationship was used by Heffernan et al (1982a) to obtain rough estimates of systolic and diastolic pressures. The relationship between pulse pressure (P_{ulp}) and stroke volume is shown below :

$$P_{ulP} = 40 \cdot \frac{SV}{SV0} \qquad (mmHg) \qquad (2.30)$$

where SV0 is the stroke volume in the unanaesthetised state. Systolic blood pressure (S_{ysP}) is calculated as :

$$S_{ysP} = MABP + \frac{2}{3} \cdot P_{ulp} \qquad (2.31)$$

and diastolic blood pressure (D_{iaP}) is calculated as :

$$D_{iaP} = MABP - \frac{1}{3} \cdot P_{ulp} \qquad (2.32)$$

2.5 THE BREATHING CIRCUIT SUB-MODEL.

Consider the schematic diagram of a semi-closed circle breathing circuit shown in figure 2.10. There are a number of factors that will affect the concentration of anaesthetic vapor inside the circuit, thus influencing the inspired concentration supplied to the patient.

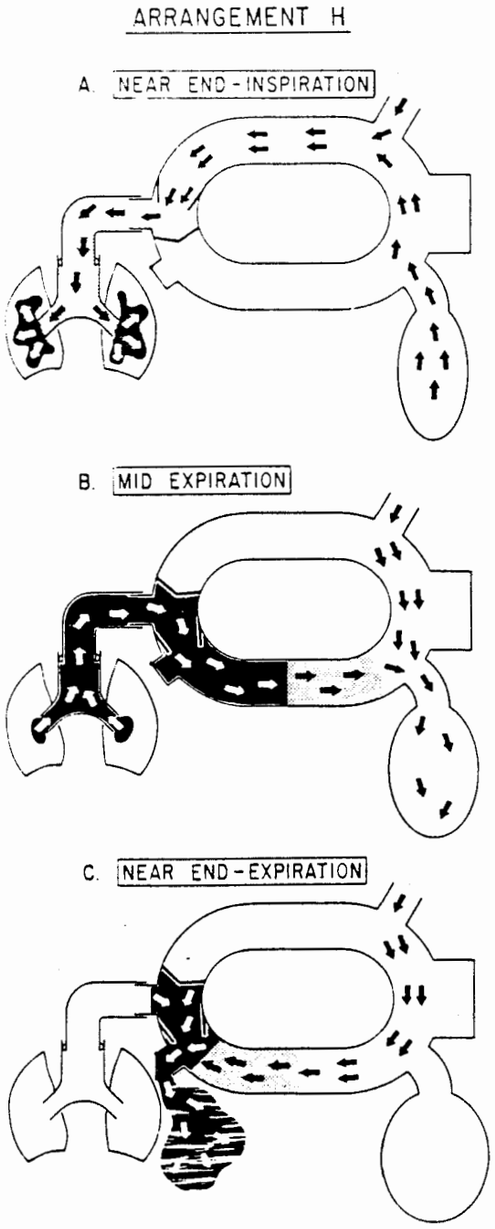
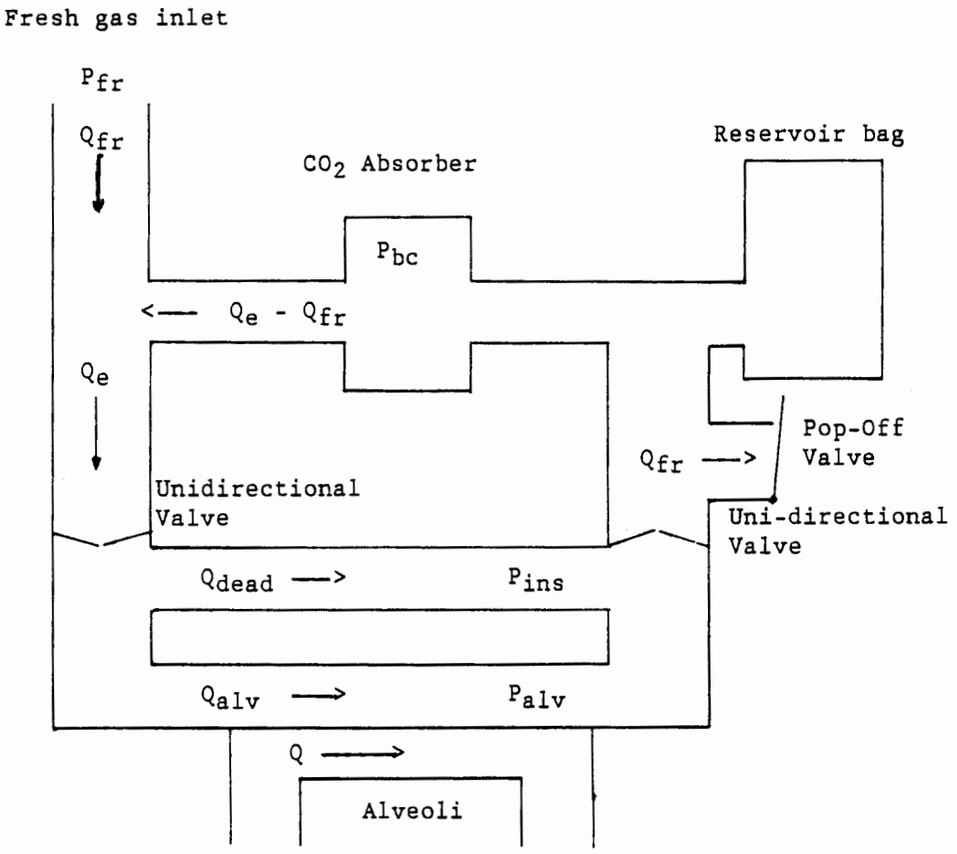


Figure 2.10 A flow diagram of the circle breathing circuit. Clear areas with black arrows are fresh gas, black areas with white arrows are alveolar gas, stippled areas with black arrows are dead space gas. Arrows indicate the movement of gas; their omision indicates no gas movement. (Reproduced from Eger 1974).



P_{alv} - Partial pressure of anaesthetic in alveoli
P_{bc} - Partial pressure of anaesthetic in CO_2 absorber
P_{fr} - Partial pressure of anaesthetic in fresh entering the circuit
P_{ins} - Partial pressure of anaesthetic in inspired air
Q_{alv} - Alveolar ventilation rate
Q_{dead} - Dead space ventilation rate
Q_e - Total ventilation rate
Q_{fr} - Fresh gas flow rate

Figure 2.11 A schematic block diagram of the semi-closed breathing circuit.

The following simplifying assumptions are made :

1/ There is no uptake of anaesthetic gas by the rubber of the breathing circuit, or by the carbon dioxide absorber.

2/ The rate of gas escaping from the pop-off valve is equal to the fresh inflow rate. (In reality oxygen is being absorbed at about 200ml/min and carbon dioxide is being

released at about 250ml/min thus changing the output to input ratio).

3/ Ventilation is considered to be non-pulsatile as the equilibration times are large compared to the ventilation cycle.

4/ Ventilation is assumed to be assisted by a ventilator and is not affected by halothane.

There are two main possibilities for gas flow rate combinations within the circuit :

A/ The fresh gas flow rate (Q_{fr}) is greater than the total minute ventilation rate (Q_e).

B/ The fresh gas flow rate is less than the total minute ventilation rate.

A/ When Q_{fr} is greater than Q_e the equation relating inspired anaesthetic concentration is simply :

$$P_{ins} = P_{fresh} \qquad (2.33)$$

The large fresh gas inflow flows through the breathing circuit and expels all expired alveolar and dead space gas through the pop-off valve thus preventing it from mixing in the breathing circuit. A certain amount of fresh gas is also lost.

B/ When Q_{fr} is less than Q_e the amount of gas flowing into the respiratory tract is the sum of the amount of gas flowing into the circuit and the amount of gas flowing back from the CO_2 absorber.

$$Q_e.P_{ins} = Q_{fr}.P_{fr} + (Q_e - Q_{fr}).P_{bc} \qquad (2.34)$$

which simplifies to

$$P_{ins} = \frac{Q_{fr}}{Q_e}.P_{fr} + \frac{(Q_e - Q_{fr})}{Q_e}.P_{bc} \qquad (2.35)$$

The concentration of anaesthetic in the absorber P_{bc} is the time integral of the amount flowing into the absorber minus the amount flowing out of the absorber all divided by the volume of the absorber. There are two gas flow rate combinations to consider when defining an equation for the concentration of gas in the absorber (amount = flow rate times concentration.) :

i/ The alveolar ventilation rate (Q_{alv}) is less than the fresh gas inflow rate (Q_{fr}).

ii/ The alveolar ventilation rate is greater than the fresh gas inflow rate.

i/ When Q_{alv} is less than Q_{fr} then the equation for the concentration of anaesthetic in the absorber is :

$$P_{bc} = \frac{1}{V_{bc}}.\int_0^t \underbrace{[(Q_e - Q_{fr}).P_{ins}}_{\text{(Amount in)}} - \underbrace{(Q_e - Q_{fr}).P_{bc}]}_{\text{(Amount out)}}dt \qquad (2.36)$$

The gas flowing into the absorber consists of expired dead space gas which is at the inspired concentration, as the expired alveolar gas is all lost through the pop-off valve.

ii/ When Q_{fr} is less than Q_{alv} not all the expired alveolar gas is lost from the system. Thus the equation for the concentration of anaesthetic in the absorber becomes :

$$P_{bc} = \frac{1}{V_{bc}} \cdot \int_0^t [(Q_{dead} \cdot P_{ins} + Q_{alv} \cdot P_{alv} - Q_{fr} \cdot P_{alv}) - (Q_e - Q_{fr}) \cdot P_{bc}] \cdot dt$$

(Amount in) - (Amount out)

(2.37)

After simplification the differential equations of equation 2.36 and 2.37 become :

i/ $Q_{alv} < Q_{fr}$

$$dP_{bc}/dt = \frac{1}{V_{bc}} \cdot (Q_e - Q_{fr}) \cdot (P_{ins} - P_{fr}) \qquad (2.38)$$

ii/ $Q_{fr} < Q_{alv}$

$$dP_{bc}/dt = \frac{1}{V_{bc}} \cdot [Q_{dead} \cdot (P_{ins} - P_{bc}) + (Q_{alv} - Q_{fr}) \cdot (P_{alv} - P_{fr})] \qquad (2.39)$$

SUMMARY

In this section a number of differential equations have been defined that describe a generalised n compartment multiple model of the uptake and distribution of an anaesthetic gas. Before these equations can be used a decision has to be made on the number of compartments to use, and which tissue groups will make up each compartment. The next chapter discusses aspects pertaining to decisions such as these. In addition a method whereby these equations can be implemented on a digital computer is described.

CHAPTER 3

COMPUTATIONAL ASPECTS OF THE MODEL

CHAPTER 3

COMPUTATIONAL ASPECTS OF THE MODEL.

In the previous chapter a generalised multiple model of the uptake and distribution of halothane was described. Before this model can be put to practical use a decision has to be made about the number of compartments required and the tissue types that each compartment will consist of. Data such as compartment tissue volumes and anaesthetic gas partition coefficients for each compartment have to be obtained. The equations resulting from this new model may then be used in a computer program subroutine for simulation purposes. This chapter describes the computational method. The question of the availability of quantitative data for the model constants is addressed. In addition a method for calculating the accuracy of the simulation using a digital computer is described.

3.1 CHOICE OF COMPARTMENTS AND QUANTITATIVE DATA.

A compartment consists of a group of tissues with similar blood flows, anaesthetic gas solubilities and capacities. A number of problems arise when a selection of model compartments is made. A model with a compartment for each tissue type in the body would presumably give more realistic results than a model with say 4 compartments, each consisting of a group of tissues with similar characteristics. However the computation required for a complex model would slow the simulation down considerably, while the quantitative data required for the compartment equation constants are often not

available and would have to be acquired through animal and human studies.

Numerous models of the uptake and distribution of anaesthetic agents have been described since 1950. It was decided to combine features of several existing models for the initial simulator program. A demonstration model was created which consists of three separate sub-models viz, an uptake and distribution sub-model consisting of 12 body compartments after Zwart et al (1972), a cardiovascular sub-model based on the model described by Zwart et al (1972), and a breathing circuit sub-model after Heffernan (1984). The demonstration model compartments are as follows :

1/ Arterial compartment - Left heart, aorta, and arteries.
2/ Brain grey matter.
3/ Brain white matter.
4/ Heart muscle supplied by the coronary arteries.
5/ Well perfused organs - kidneys, adrenals, thyroid.
6/ Poorly perfused tissues - red marrow and non-fatty subcutaneous tissue.
7/ Fat - Fat and fatty marrow.
8/ Splanchnic organs - those drained by the portal and hepatic circulation.
9/ Skeletal muscle - muscle and skin nutritive.
10/ Skin shunt.
11/ Venous - Vena cava and right heart.
12/ Lung.

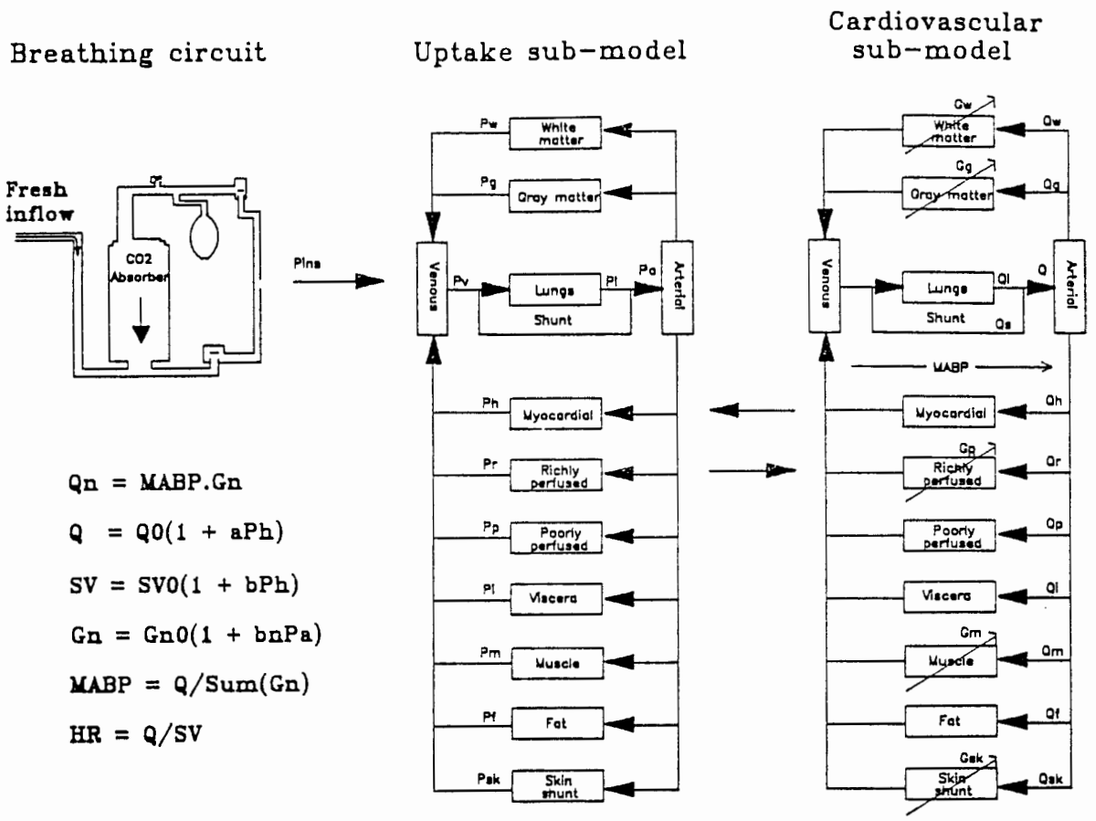
The cardiovascular sub-model used was basically that of Zwart et al (1972) except for the relationship of the stroke volume

to halothane and the stroke volume-pulse pressure estimation, which are described in the previous chapter.

The breathing circuit sub-model is described in detail in the previous chapter.

The values for tissue volumes, blood volumes, blood flow rates, blood and tissue gas partition coefficients, breathing circuit volumes, and lung compartment dimensions are all after Smith et al (1972). In 5 of the cardiovascular sub-model compartments the conductance is assumed to vary linearly with respect to the arterial halothane concentration. The values for the constants in the conductance equations were calculated using the data given by Smith et al. The constants and the formula's used to calculate them are listed in APPENDIX A.

Figure 3.1 shows a schematic diagram of the demonstration multiple model. A list of the model parameter abbreviations and the subscripts used to distinguish between the different model compartments is given in APPENDIX B



Pn	= nth compartment halothane partial pressure	(Frac atm)
Qn	= Blood flow through the nth compartment	(l/min)
Gn	= Conductance of the nth compartment	(l/min/mmHg)
MABP	= Mean arterial blood pressure	(mmHg)
HR	= Heart rate	(Beats/min)
SV	= Stroke volume	(l)
SV0	= stroke volume in the unanaesthetised state	(l)
Q	= Cardiac output	(l/min)
Q0	= Cardiac output in the unanaesthetised state	(l/min)
Ph	= Heart muscle halothane partial pressure	(Frac atm)
Pa	= Arterial halothane partial pressure	(Frac atm)
a,b, bn	= Constants	

Figure 3.1 A schematic diagram of the demonstration model. The cardiovascular model is concerned with blood flow to the various compartments, the uptake model is concerned with the concentration of anaesthetic in the various compartments. The large arrows between the two models indicate that they influence one another. All other arrows indicate the direction of gas and blood flows. The semi-closed breathing circuit model determines the concentration of anaesthetic gas being inspired. Diagonal arrows on compartments in the cardiovascular model indicate variable conductance's. All other compartments have constant conductances. The breathing circuit air flow is controlled by a ventilator (not shown).

3.2 EULER INTEGRATION.

It is fairly easy to build an analogue computer circuit that can perform an integration (Bellville and Hara 1966). To integrate a function on a digital computer requires the use of special iterative techniques. The simplest and fastest method, though not the most accurate, is known as Euler integration. A detailed description of the method is given in APPENDIX C.

Consider the concentration of an anaesthetic agent in a general compartment n as an example (equation 2.10). The derivative of $P_n(t)$ with respect to time is :

$$dP_n/dt = K_n \cdot Q_n \cdot [P_a - P_n] \qquad (3.1)$$

where K_n is a constant determined by the tissue and blood volumes and solubilities of that compartment and Qn is the blood flow rate through the compartment. The time interval over which one wishes to know the concentration is then divided into small equal increments of size delta_t. To calculate the value of P_n after time interval delta_t the Euler equation (equation C.7) is used :

$$P_n(t + \text{delta\_t}) = P_n(t) + \text{delta\_t} \cdot dP_n/\text{delta\_t} \qquad (3.2)$$

Here $P_n(t)$ is the initial value of P_n and $P_n(t + \text{delta\_t})$ is the value of P_n after time delta_t. Before the next iteration takes place $P_n(t)$ is set equal to $P_n(t + \text{delta\_t})$ as this is now the new initial (current) value of P_n.

In a pascal program this would be written as :

```
dPndt := Kn * Qn * [Pa - Pn];
Pn := Pn + delta_t * dPndt;
```

The sign := means "is set equal to", and '*' indicates a multiplication. Thus to simulate the way in which P_n changes with time the above two equations would be repeatedly calculated over the time interval delta_t and the new value of P_n displayed after each iteration. The smaller delta_t is made, the more accurate the results will be as the accuracy of Euler integration is proportional to the step size.

If one considers that the blood flow through the compartment n also changes with time then (using equation 2.27) the pascal program lines for compartment n becomes :

```
Qn := MABP * Gn;
dPndt := Kn * Qn * [Pa - Pn];
Pn := Pn + delta_t * dPndt;
```

The conductance G_n is a function of the concentration of halothane in the arteries P_a and the pascal code must calculate this change accordingly :

```
Gn := Gn0 * ( 1 + bn * Pa);
                                  Cardiovascular
Qn := MABP * Gn;                  sub-model.

dPndt := Kn * Qn * [Pa - Pn];
                                  Uptake sub-
Pn := Pn + delta_t * dPndt;       model.
```

A set of equations such as this must be executed for each of the compartments of the model for every time interval delta_t. At the end of this set of equations the cardiac output, arterial blood pressure, and heart rate equations have to be executed to calculate the changes in these variables during the time interval delta_t. Thus

computer code for the equations from the three sub-models are grouped in the order in which they are to be calculated in a model subroutine so that they may all be executed by a single subroutine call elsewhere in the program. A full listing of the pascal model subroutine is given in APPENDIX D.

3.3 THE ACCURACY OF EULER INTEGRATION

In appendix C it was stated that the error due to Euler integration is proportional to the step size dt over which each iteration takes place. Reducing the step size will decrease the error but will increase the number of iterations required over a set time interval thus slowing the simulation down. An additional problem is created if the simulator is designed so that variable simulation speeds are possible. ie, at 1X real time the simulator graphics will be updated after every second of simulated time and at 60X real time the graphics will be updated after every 60 seconds of simulated time. This results in different step sizes with their correspondingly different errors.

Updating a computer graphics screen takes up a large amount of time compared to the time taken for one iteration of a model's subroutine. A method of reducing the error without slowing the simulator down by very much is to break up the time interval between graphics updates (no matter what speed is selected) into a number of smaller intervals and to iterate the model a number of times between graphics updates.

To calculate the error due to Euler integration a

differential equation with an exact solution is required. The simulated results using this equation can then be compared to the exact solution and the difference (or error) between then calculated. The exact solution of the differential equation describing the uptake of anaesthetic by any compartment can be obtained if the blood flow through the compartment is assumed to be constant. The equation for the concentration of anaesthetic in a compartment is given by equation 2.10 and is:

$$P_n(t) = P_n(0) + K_n \cdot Q_n \cdot \int_0^t [P_a - P_n]dt \qquad (3.3)$$

Assuming constant blood flow (Q_n) through the compartment, the exact solution for this equation can be obtained. Take the time derivative of equation 3.3 to obtain :

$$\frac{dP_n(t)}{dt} = K_n \cdot Q_n \cdot [P_a - P_n(t)] = -K_n \cdot Q_n \cdot [P_n(t) - P_a] \qquad (3.4)$$

Equation 3.4 can be rearranged as :

$$\frac{dP_n(t)}{P_n(t) - P_{na}} = -K_n \cdot Q_n \cdot dt \qquad (3.5)$$

Substitute U for $P_n(t)$ and V for t and integrate both sides of equation 3.5 :

$$\int_{P_n(0)}^{P_n(t)} \frac{dU}{U - P_{na}} = -K_n \cdot Q_n \cdot \int_0^t dV \qquad (3.6)$$

which becomes :

$$[L_n(U - P_{na})]_{P_n(0)}^{P_n(t)} = -K_n \cdot Q_n [V]_0^t \qquad (3.7)$$

Solving for $P_n(t)$ yields :

$$\frac{P_n(t) - P_{na}}{P_n(0) - P_{na}} = e^{-Q_n \cdot K_n \cdot t} \qquad (3.8)$$

or

$$P_n(t) = P_{na} + (P_n(0) - P_{na}) \cdot e^{-Q_n \cdot K_n \cdot t} \qquad (3.9)$$

The solution indicates that the concentration of anaesthetic in the compartment approaches the arterial concentration exponentially at a rate determined by the blood flow through the compartment, and on the constant K_n.

A test program was written to calculate the error due to Euler integration at different speeds and for different numbers of iterations between graphics updates.

It was decided to use a "fast" compartment,the cerebral grey matter compartment and a "slow" compartment, the muscle compartment for the test. The values for K_n and the blood flow rate Q_n for the differential equations describing these two compartments are given in table 3.1.

Compartment	K_n	Q_n	$K_n \cdot Q_n$
Grey matter	0.4691	0.6	0.2815
Muscle	0.0086	1.2	0.0103

TABLE 3.1 Constants for the grey matter and muscle compartments.

Note that although the blood flow to the muscle compartment is twice as large as the grey matter compartment, the muscle compartment has a larger tissue and blood volume. Thus the

constant K is very much smaller.

The partial pressure in the grey matter and muscle compartments was calculated as an actual value and an estimated value using Euler integration. The percentage error between then was calculated and plotted against a time axis. This was done for an inflow concentration of 2% halothane with 0% initial concentration, and for an inflow concentration of 0.2% halothane with a 2% initial concentration. The simulation was run at speeds of 1, 10, and 60 times real speed and with 1, 2, 4, 10, and 20 iterations between graphics updates. The results are presented in chapter 5 and a listing of the error calculation program is given in APPENDIX E. Table 3.2 lists a summary of results for these compartments.

The simulation was performed over 20 minutes for the fast compartment, and over 360 minutes for the slow one. Diagrams of these errors over these time intervals are presented in the results section in figures 5.2 to 5.9. From these results it was decided that 10 iterations between graphics updates would allow the simulator to operate at an acceptable speed while keeping the error due to the calculations less than 2%.

3.4 DISCUSION.

In this section a demonstration model was described. A technique for calculating differential equations on a digital computer, known as Euler integration, was described. The

		GREY MATTER COMPARTMENT (SIMULATION OVER 20 MINUTES)							
		0 TO 2% INPUT				2 TO 0.2% INPUT			
Speed	Iterations /step	1	4	10	20	1	4	10	20
1X	Maximum % error	0.203	0.050	0.020	0.010	0.257	0.064	0.026	0.129
	Time	37.5	48.8	71.5	109.3	-	-	-	-
10X	Maximum % error	2.090	0.511	0.204	0.102	2.593	0.644	0.257	0.129
	Time	3.9	5.1	7.3	11.0	-	-	-	-
60X	Maximum % error	14.732	3.181	1.239	0.614	16.18	3.905	1.551	0.773
	Time	0.8	0.9	1.4	2.0	-	-	-	-
MUSCLE COMPARTMENT (SIMULATION OVER 360 MINUTES)									
60X	Maximum % error	0.488	0.121	0.049	0.024	0.569	0.142	0.568	0.028
	Time	7.5	9.5	14.6	22.4	-	-	-	-

TABLE 3.2 Summary of results of the error calculation program. Times are in seconds and do not include the time taken to perform graphics updates.

errors due to this method were defined and a test program described that was used to calculate these errors. The errors are summarised in table 3.2. It was found that the errors were not only dependent on the size of the time step size, but also on the size of the derivative of the equation. ie, a fast compartment's rate of change of halothane concentration is much greater than a slow compartment causing the error in the estimate to be greater.

Another feature that the error calculation program highlighted is that the error approaches zero as the compartment concentration equilibrates with the blood

concentration of halothane. The maximum error occurs only when the concentration of halothane is changing rapidly. Figures 5.2 to 5.9 in the results section show this trend clearly.

From these results it was decided that a value of 10 iterations between graphics updates would give a maximum error of less than 2% at a simulator speed of 60 times real time. ie, a maximum step size of 0.1 minutes (or 6 seconds) gives an error of less than 2% for the grey matter compartment.

The next section describes the simulator program and the commands that are used to control it.

CHAPTER 4

THE SIMULATOR SOFTWARE ENVIROMENT

CHAPTER 4

THE SIMULATOR SOFTWARE ENVIRONMENT.

The software for the demonstration model was developed with a number of goals in mind :

I) To be educational
II) To be easy to use
III) To be transportable
IV) To allow testing and updating of the model

4.1 HARDWARE AND SOFTWARE SELECTION.

The software language chosen was TURBO PASCAL V4.0[1] implemented on an IBM personal computer. Pascal was chosen as it is more versatile than BASIC or FORTRAN on a PC, and a full library of graphics routines were available in the form of the TURBO GRAPHIX TOOLBOX[1]. In addition TURBO PASCAL V4.0 allows the programmer to compile certain sections of a program into what is termed a UNIT. Thus if more than one model is developed for testing, the model subroutines and parameter declarations can be compiled into individual units allowing the programmer to select the required model at compile time with ease.

Due to the large amount of information that needs to be displayed by the simulator, the IBM monochrome graphics card was selected over the IBM color graphics card for the first version of the program due to its greater pixel resolution.

[1]TURBO PASCAL and TURBO GRAPHIX TOOLBOX are registered trademarks of Borland International Inc. California.

4.2 DATA DISPLAY

Before the program was written the graphics screens were designed. In the demonstration model there are 60 parameters of interest that are available for displaying as a bar-graph or line-graph. In addition there are 4 parameters that can be controlled by the operator and 5 parameters that indicate the status of the simulator. It would be impractical to display all this information on the screen at the same time. Thus the operator must make a decision as to which parameters are to be displayed.

The graphics screen is divided up into 4 permanent windows, one pop-up window, and a menu/message line. Figure 4.1 shows the layout of the graphics screen without the pop-up window and figure 4.2 illustrates how the graph window covers the two right hand windows.

The top left window is the status window and displays such information as elapsed time, the simulator speed, whether the simulator is running or stopped, the amount of free memory and the name of an external dump file.

The bottom left window is the control window which displays the control parameters in bar-graph form.

The middle window displays bar-graphs of 9 fixed model parameters which include information about the cardiovascular sub-model.

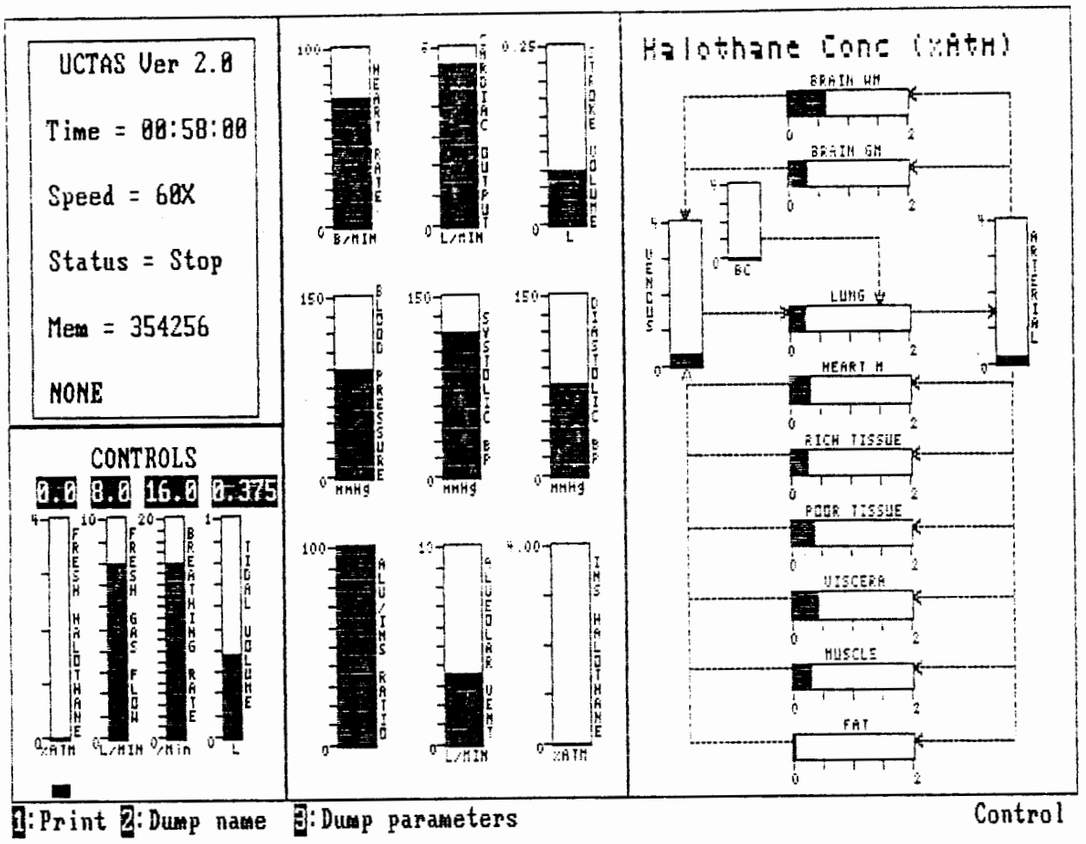


Figure 4.1 The simulator graphics screen.

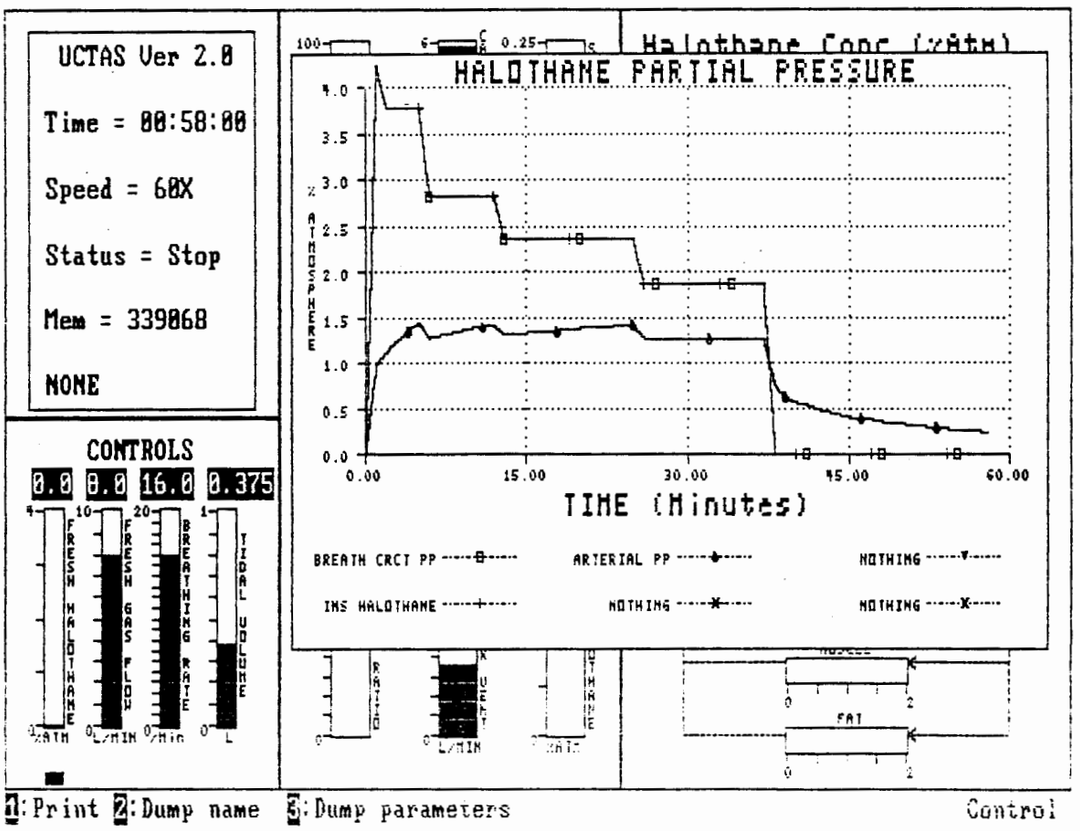


Figure 4.2 The simulator graphics screen with the pop-up graph window.

The right hand window contains 12 bar-graphs each of which represents a compartment in the model. This window can be selected to display either the concentration of halothane in the compartments, the blood flow rate through the compartments, the conductance of the compartments, or the volume of halothane vapour taken up by the compartments.

The pop-up window is the graph window and may be used to plot up to 6 parameters at a time over some selectable time scale. The selected parameters are plotted even when the graph window is not visible thus allowing trends to be viewed at any time during a simulation.

A text screen is available to display help information about various simulator commands. The graphics screen is saved before the text screen is called, then restored when the operator has finished with help. This screen (Shown in figure 4.3) consists of a menu of help topics in the top 1/3 of the screen, and the help text for the current topic displayed in the lower 2/3 of the screen. Up to 30 help topics may be viewed one at a time on this screen by selecting a topic with the cursor keys.

Two further text screens are available for the selection of parameters that are to be displayed on the graph screen or sent to the dump file. The top section of the screen lists the current parameters while the lower part of the screen lists the parameters that can be selected (Figure 4.4 and 4.5). A parameter is selected from the top list with the cursor keys and the enter key pressed. This sends the cursor indicator to the lower list. The cursor keys are used to

select a parameter. If the return key is pressed then the display parameter in the top list is updated. If the escape key is pressed then the display parameter in the top list is set to display or output no parameter.

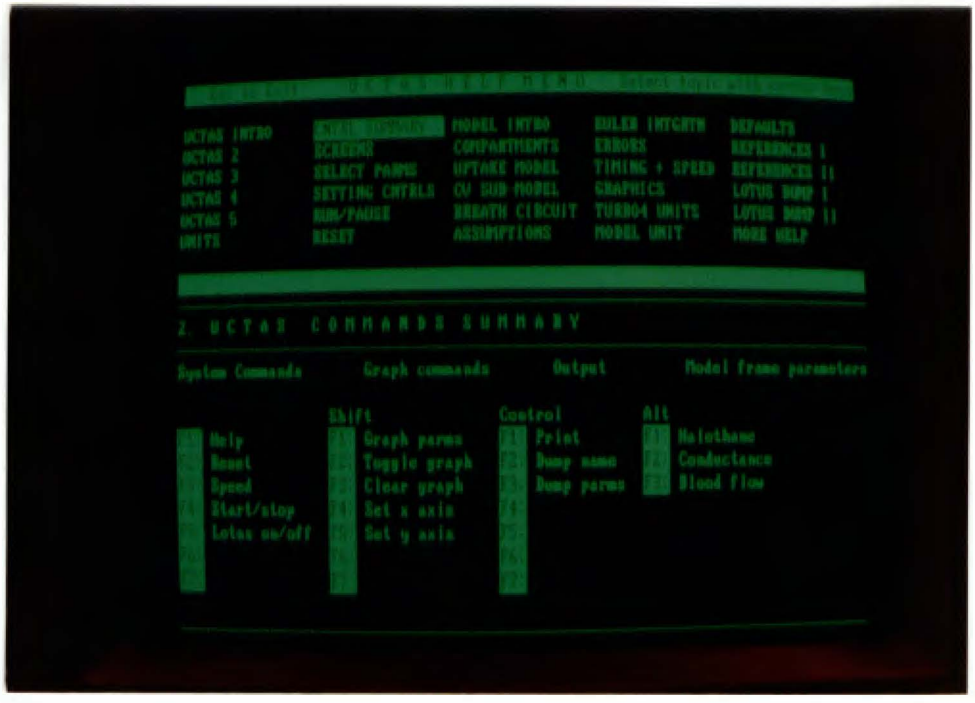


Figure 4.3 The help screen.

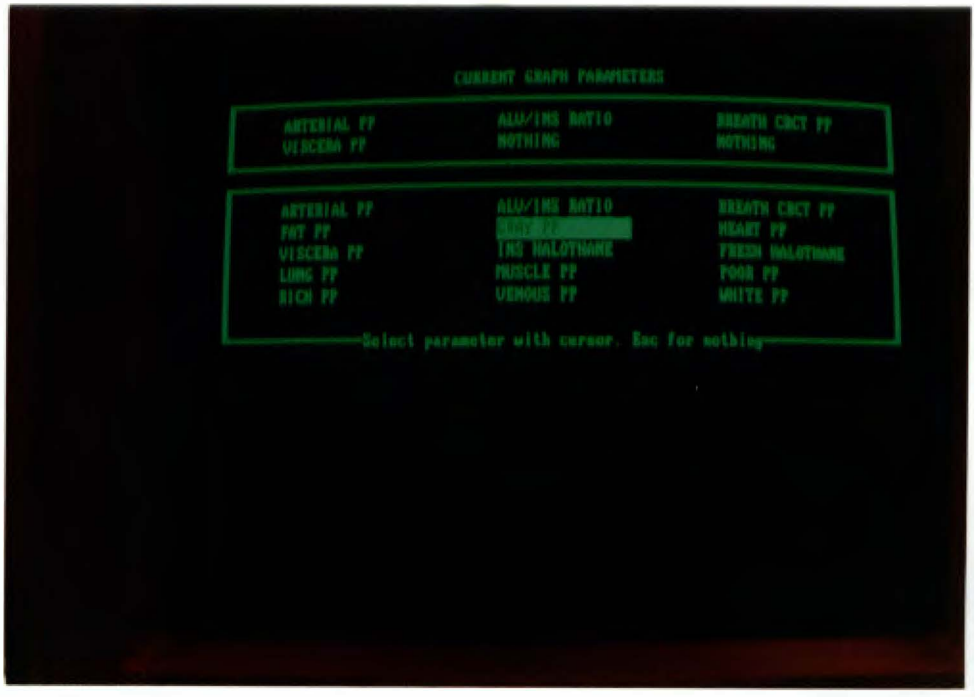


Figure 4.4 Graph parameter selection screen.

Figure 4.5 Dump file selection screen.

4.3 THE MAIN PROGRAM OUTLINE

Figure 4.6 shows a flowchart of the simulator program. A full listing of the program is given in APPENDIX G. The listing contains comment lines that explain the function of each subroutine. The main program is at the end of the listing. A description of the various pre-compiled utility units that are called throughout the program is given in APPENDIX F.

Normally the graphics card will be in text mode when the program is executed. Before switching to the graphics mode a check is made to determine if the correct graphics card exists in the machine, and if enough memory is available for the graphics system to use. If any fault is found the operator is informed of the error and the program terminates. The help screens are then loaded and the default window,

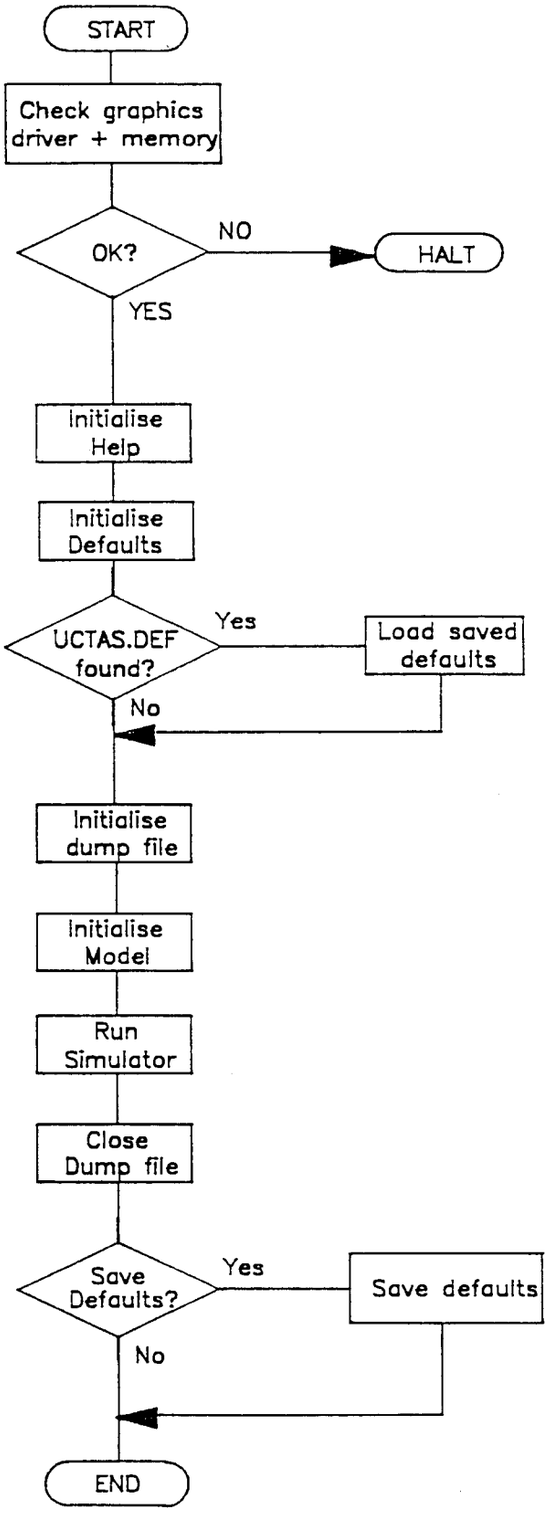


Figure 4.6 The main simulator flowchart.

graph, and dump file parameters are initialised. If the saved defaults file is found, then these default parameters are loaded over the previous ones. The dump file is then opened and the model constants are initialised. The program then transfers control over to the simulator subroutine until the operator wishes to end the session and return to the operating system. When this occurs the dump file is closed and the operator can save the currently selected window, graph and dump file parameters in the default file before the program is terminated.

4.4 THE SIMULATOR SUBROUTINE

Figure 4.7 shows the flowchart of the simulator subroutine. The first operation that is performed is to enter graphics mode. TURBO GRAPHICS TOOLBOX allows one to define graphics windows and to define a world coordinate system within each window. In addition the toolbox makes two screens available for drawing graphics ; the visible screen, and the invisible memory screen. Thus all drawing is done in the invisible screen and copied over to the visible screen once the drawing operations are completed. Once the windows and worlds have been defined the bar-graph frames are drawn in their correct places, the status window is initialised, and the control window bar-graphs are updated to their default values. The pop-up graphs axes are drawn and the graph window saved on the window stack for future retrieval. The subroutine then executes code in a loop until the exit option is chosen. Within the loop the following operations take place :

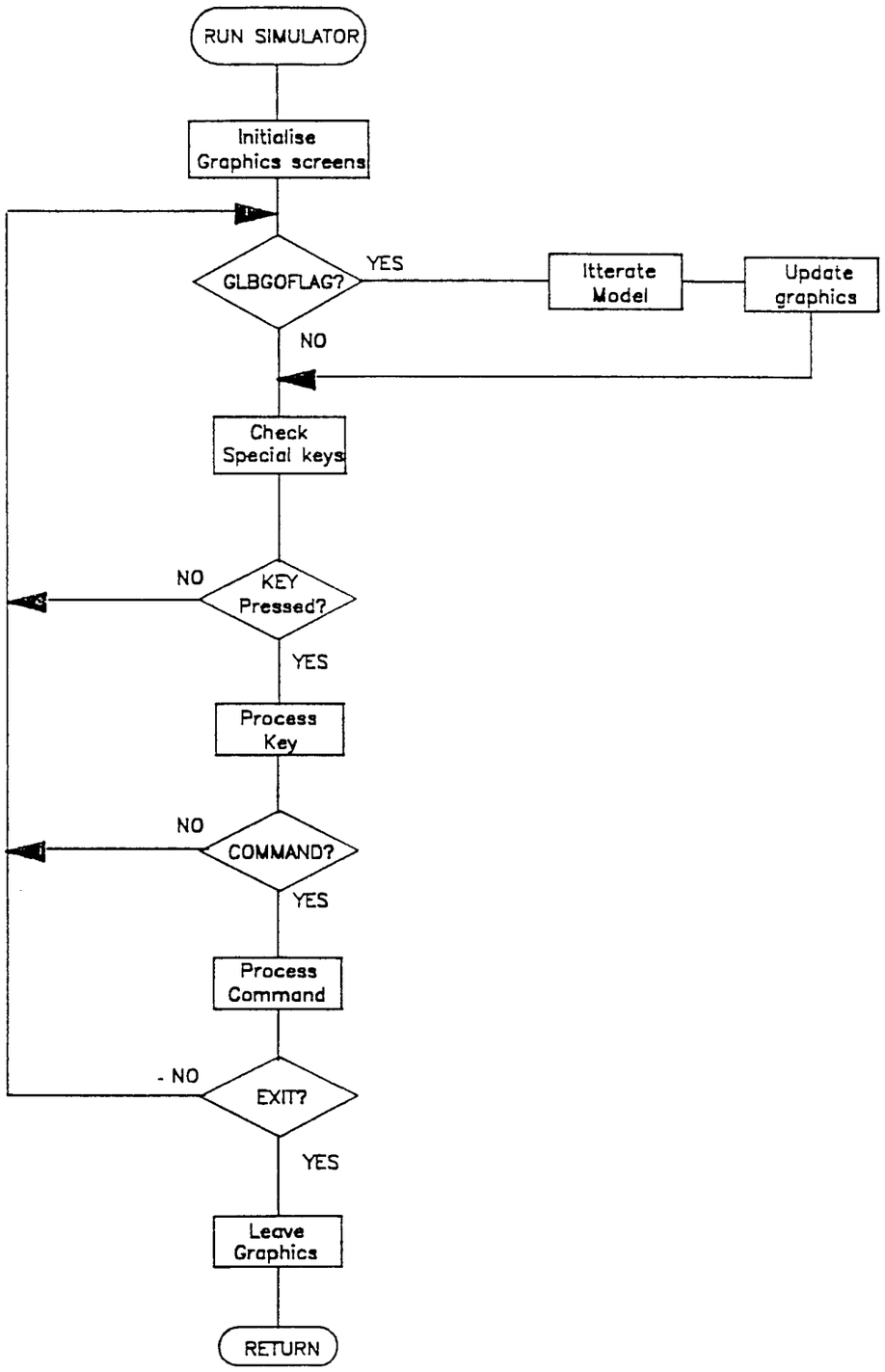


Figure 4.7 The simulator subroutine flowchart.

1/ If the run flag is true then the model subroutine is executed 10 times over a time interval specified by the current simulator speed. The current graphics display is then updated.

2/ A check is made to see if one of the special keys (Shift, Alt or Control key) has been pressed. If so then the menu line corresponding to that key is written at the bottom of the screen.

3/ If any other key was pressed then the program executes a CASE statement to determine if that key corresponds to a command.

4/ If a command key has been pressed then the command is processed by calling the appropriate subroutines from within the CASE statement.

Once the exit option has been chosen the operator is asked to confirm the choice before the graphics screen is cleared and text mode restored.

4.5 THE MODEL ITERATION SUBROUTINE.

The simulator can operate at speeds of 1, 5, 15, 30, 60 and 600 times real time. ie the interval in seconds between graphics output during the simulation is equal to the speed of the simulator. Two parameter parameters are defined for use in the iteration subroutine. *Iteratetime* is the number of

seconds over which the 10 iterations of the model equations will take place and *delta_t* is the time interval in minutes over which each single iteration will take place. Delta_t is measured in minutes as the flow rate constants used in the model subroutine differential equations are given as volume per minute. Figure 4.8 shows the flowchart of the model iteration subroutine. The subroutine executes the model equations 10 times then updates the elapsed time counter before returning to the calling routine.

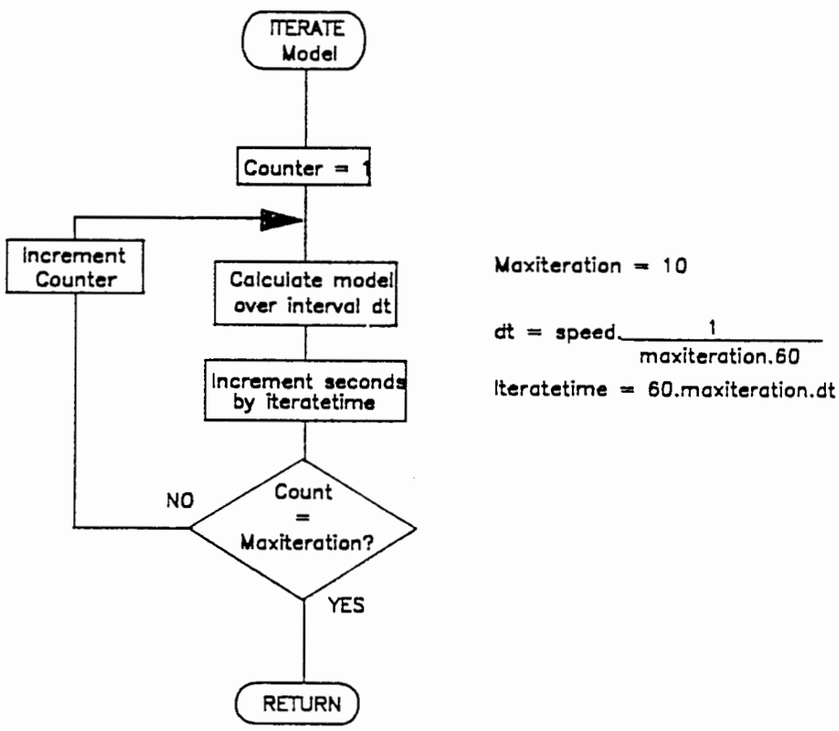


Figure 4.8 The model iteration subroutine flow chart.

4.6 OPERATION OF THE SIMULATOR

All commands to the simulator are entered via the keyboard using the function keys and cursor keys. The IBM PC has 10 function keys on the keyboard. These may be used on their own or in conjunction with the control, alt, or shift keys. This gives the operator a selection of 40 function keys with which the simulator may be controlled. The menu line only has space to display 80 characters. It would be impractical to have a menu with all possible function key options displayed, thus only 1/4 of the options are displayed at a time. By holding down the control, alt or shift key on its own, the menu line corresponding to those 10 function keys will be displayed.

The cursor keys are used to move a selection bar around on the screen to select topics in the help screen, to select parameters in one of the parameter selection screens, and to select and adjust control parameters in the control window.

Other keys that are used are the return or enter key and the escape key. For certain commands a filename or number will have to be entered using the alpha-numeric keys terminated by pressing the return key.

4.6.1 STARTING THE SIMULATOR.

The simulator hardware requirements are :

- IBM PC or compatible.
- Monochrome screen (Hercules graphics card or equivalent).
- 512K of memory
- One floppy disk drive
- Epson graphics printer optional

The program disk must have the following files on it :

UCTAS.EXE - Main program
UCTAS.HLP - Help file
ERROR.MSG - Graphics error messages
4X6.FON - Graphics small font
12X9.FON - Graphics large font

OPTIONAL

UCTAS.DEF - Default screens file.

To run the program, place the disk in the current drive and type the program name UCTAS, followed by return. The program takes a while to set up the graphics screens and will issue warnings if the graphics hardware is not found, or if there is not enough memory available. If the file UCTAS.DEF is not found on the disk the initial graphics display will look like that shown in figure 4.1. The default graph parameters are arterial halothane concentration, inspired concentration, and breathing circuit concentration. The default menu is displayed on the bottom line. The following sections will discuss the available commands and how to run a simulation.

4.6.2 THE HELP FACILITY.

The help facility may be called up at any time during a simulation by pressing the F1 key. If the simulator is running at the time, the process is halted, the graphics display is saved, and the help menu displayed. The selection bar may be used to select a help topic in the top menu by moving it with the cursor keys. The corresponding help text will then be displayed in the lower part of the screen. The escape key is pressed to return to the simulator. The graphics screen is restored, and the simulation continues from where it was interrupted.

4.6.3 SETTING THE CONTROLS.

In the lower left window there are 4 control parameters. The operator can control the fresh gas concentration of halothane, the rate at which fresh gas flows into the breathing circuit, and the tidal volume and breathing rate ventilator settings. The currently active control is indicated by a small square underneath the bar-graph. This square may be moved using the left and right cursor keys. To adjust the active control use the pageup and pagedown keys for coarse control, and the up and down cursor keys for fine control. The actual value of the control parameter is indicated in numerical form at the top of each bar-graph.

4.6.4 SETTING UP THE DISPLAYS.

Use Alt F1 to Alt F4 to select the right hand window compartment parameters for display on the bar-graphs there. If a line graph of certain parameters is required then pressing F2 will pop the graph window up over the other windows. Pressing F2 again will put the graph away. Press shift F1 to select the graph parameters. The graphics screen is saved and menu of 4 groups of parameters is displayed. These parameters are grouped so that the range of the y axis of the graph will be sufficient to display all of them legibly. Select one of the groups by moving the selection bar with the cursor keys. Press return for the selection or escape to return to the simulator.

The top part of the screen now displays the current graph parameters (from 1 to 6 parameters may be graphed at a time, each indicated by a different line marker and labeled below the graph) and a list of available parameters at the bottom. Use the cursor keys to select a graph parameter that is required to be changed and press return. The selection bar moves to the lower list of parameters. Use the cursor keys to select a parameter and press return. If escape is pressed then no parameter is selected, (indicated by NOTHING in the graph parameter menu) and nothing will be plotted on the graph for that particular line type. Press escape to return to the simulator. There will be a slight delay as the new graph axis is drawn. The y axis will automatically be scaled to cover the range of the parameter group selected. The old graph will be cleared.

4.6.5 RUNNING A SIMULATION.

Once the displays have been set up and the controls initialised the simulator may be run by pressing F4. The elapsed time is displayed in the status window. The speed at which the simulation will take place is shown in the status window. The value of the speed indicator is the step size, in seconds, between iterations and graphics updates. Consider a 2 hour long anaesthetic procedure simulation as an example : at a speed of 1X the model is iterated over a time interval of 1 second. Thus to simulate the whole 2 hours will require 7200 iterations of the model; at a speed of 60X the model will be iterated over a 60 second time interval which will only require 120 iterations to simulate 2 hours of data.

The speed may be adjusted with the F3 key. The simulator may be paused with the F4 key at any time to allow the controls to be adjusted during a simulation (They may be adjusted while the simulator is running but there is a delay due to the displays being updated).

The F2 key may be used to reset the simulator. The elapsed time is reset to zero, and the controls are reset to there default values. The graph window is not cleared thus allowing multiple simulations to be run for comparison of the effects of different control inputs. Shift F3 will clear the graph if required. Shift F4 can be used to set the time axis of the graph. The default is 3 hours or 180 minutes.

4.6.6 THE LOTUS DUMPFILE.

A facility has been included in the program to give the operator the ability to dump up to 13 different parameters to a text file at the same rate that the display is updated. This data may be used to generate high quality plots of the data generated by the simulator. The text file has been designed to be compatible with [1]LOTUS 1-2-3, a spreadsheet program that can import the dump file contents and be used to produce graphs of the data. These may then be plotted on various printers and plotters.
Control F3 calls up the lotus dump file parameter selection menu. These parameters are selected in the same manner that the graph parameters are selected.

Control F2 prompts the operator for a dump file name. The old dump file is closed and the operator enters the new one. If the new file already exists the operator has the option of overwriting it or of entering the file name again. Note that the file name will have to have the extension .PRN before it can be included into a LOTUS worksheet.

F5 will toggle the dump option on or off. When it is active the current dump file name will be displayed in the status window else the word NONE will be shown. Parameters are only dumped when the simulator is running.

Control F1 will cause the current graphics screen to be printed out on an epson graphics printer or equivalent.

[1] LOTUS 1-2-3 is the registered trademark of Lotus Development Corporation

CHAPTER 5

RESULTS

CHAPTER 5

RESULTS

The results presented in this section come from two sources: data generated by the test program to determine the errors due to Euler integration, and data generated by the simulator program. In both cases the data was generated as an ASCII text file on a floppy disk. This data was then imported into a LOTUS 1-2-3 spreadsheet. Rough graphics pictures were then generated using this data and saved on disk. These pictures were then imported into LOTUS FREELANCE, a graphics package that runs on an IBM PC. The pictures were then tidied up and plotted on an HP 7475A 6 pen plotter. The resulting graphs were then reduced in size on a KONIKA ubix 180Z photocopying machine.

5.1 RESULTS OF ERROR CALCULATION PROGRAM.

As mentioned in chapter 3, the error program was run for both the muscle and the grey matter compartments at speeds of 1, 10, and 60 times real time. The program was run for both an increase and a decrease in the concentration of halothane entering each compartment. Figure 5.1 shows the resulting curves generated by both the simulated compartment equation and the exact solutions equation for the grey matter compartment at a speed of 60 times real time. The grey matter compartments errors were calculated over a simulated time period of 20 minutes while the errors in the muscle compartment were calculated over a period of 360 minutes.

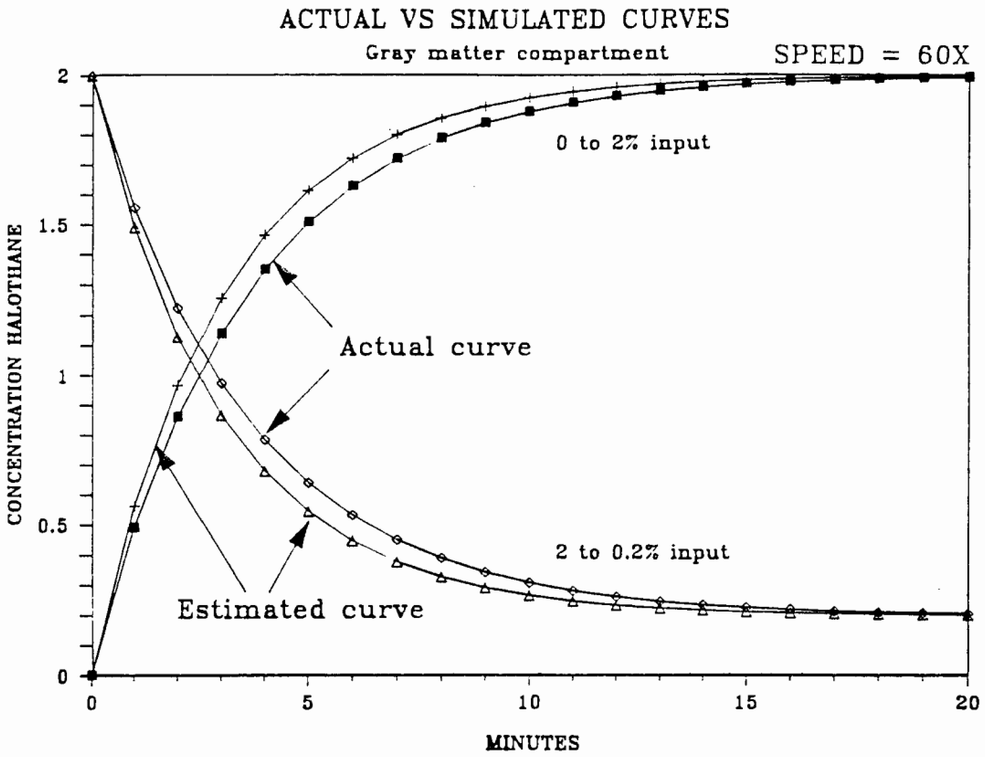


Figure 5.1 Estimated versus actual values generated by the error calculation program. In both cases the compartment tissue halothane partial pressures are assumed to be in equilibrium with the partial pressure in the blood flowing through the compartment. In the first case the blood partial pressure is changed from an equilibrium value of 0% to 2%. In the second case the blood partial pressure is changed from an equilibrium value of 2% to 0.2%. In both cases the partial pressure in the compartment eventually equilibrate with the blood. The difference between the actual and the estimated curve in both cases is the error due to the estimation.

Figures 5.2 and 5.3 are plots of the cumulative error in the grey matter compartment at a speed of 1 times real time for a 0 to 2% and a 2 to 0.2% step change of the halothane partial pressure in the blood flowing into the compartment. The 4 curves on the graphs correspond to 1, 4, 10, and 20 iterations of the differential equation between each step, ie, the time interval between data outputs is divided up into 1, 4, 10, or 20 equal sized segments.

Figures 5.4 and 5.5 are similar to the previous 2 figures except that these simulations were performed at a speed of 10 times real time. Similarly figures 5.6 and 5.7 show the cumulative errors at a speed of 60 times real time.

Figures 5.8 and 5.9 show the cumulative errors for the muscle compartment at a speed of 60 times real time. No further simulations are displayed for this compartment as the resulting errors were an order of magnitude less than the faster grey matter compartment.

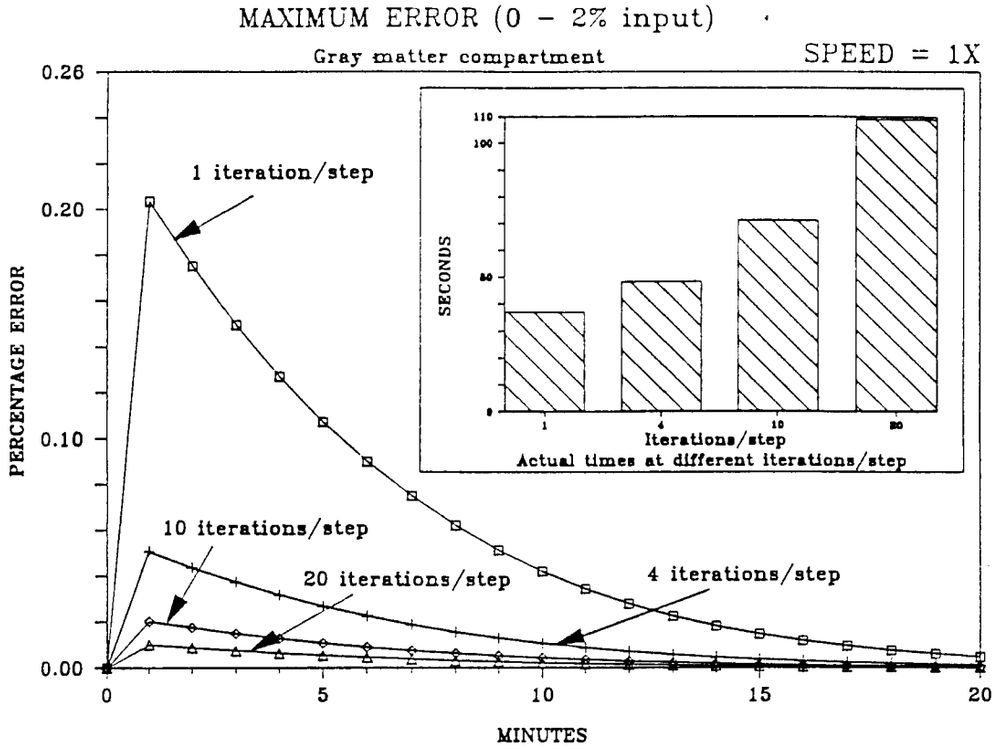


Figure 5.2 Error in a "fast" compartment at a speed of 1 times real time for a 0 to 2% step in input partial pressure. The simulation was repeated for 1, 4, 10, and 20 iterations between graphics updates. The inset bargraphs indicate the actual time in seconds that the computer took to generate 20 minutes of simulated data.

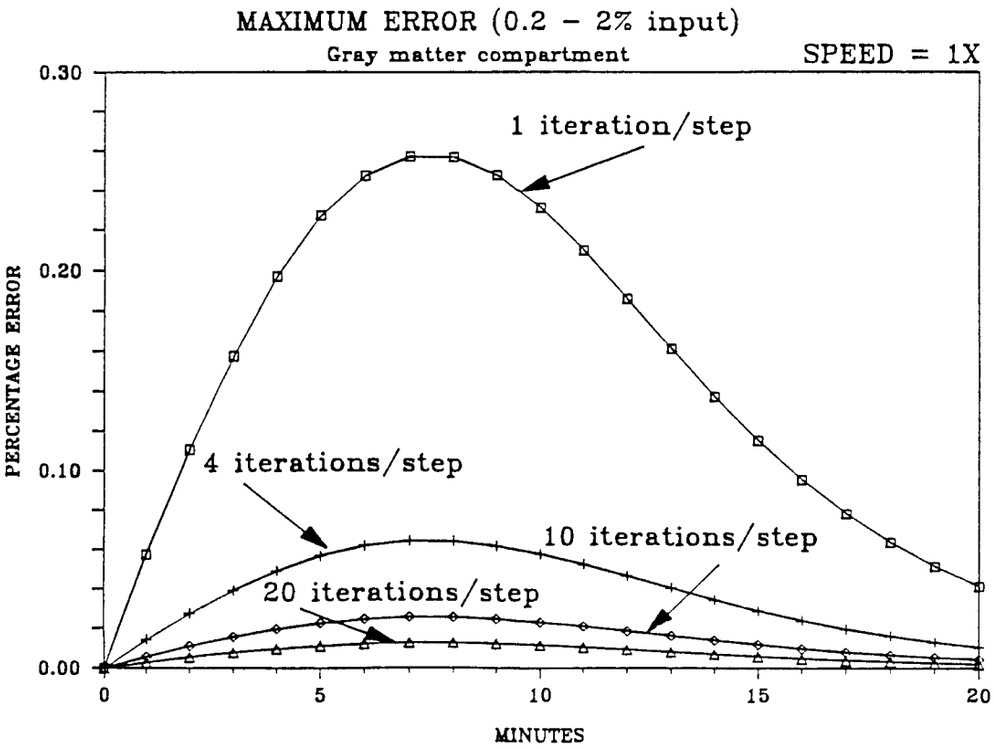


Figure 5.3 Error in a fast compartment at a speed of 1 times real time for a 2 to 0.2 % step in input partial pressure.

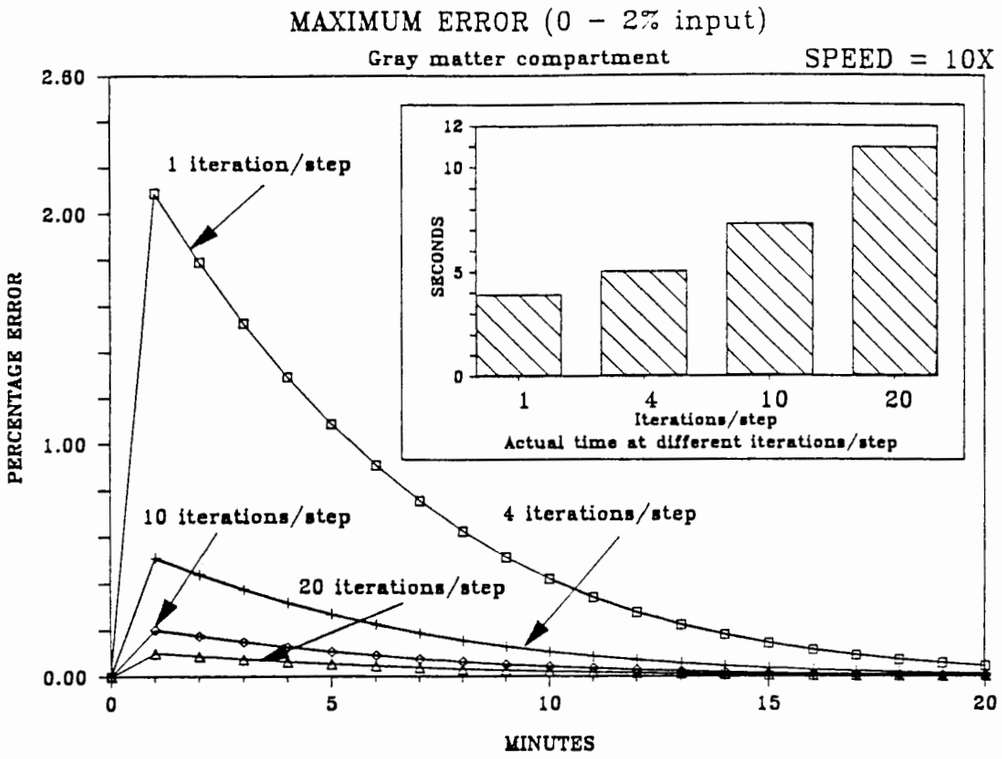


Figure 5.4 Error in a "fast" compartment at a speed of 10 times real time for a 0 to 2% step in input partial pressure. The simulation was repeated for 1, 4, 10, and 20 iterations between graphics updates. The inset bargraphs indicate the actual time in seconds that the computer took to generate 20 minutes of simulated data.

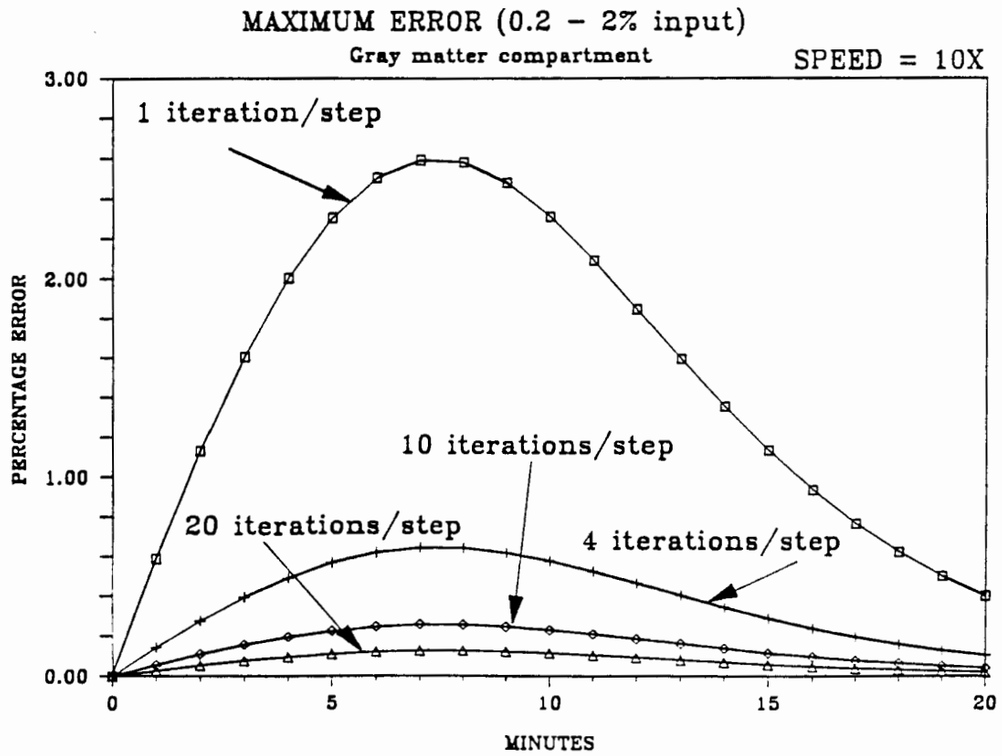


Figure 5.5 Error in a fast compartment at a speed of 10 times real time for a 2 to 0.2 % step in input partial pressure.

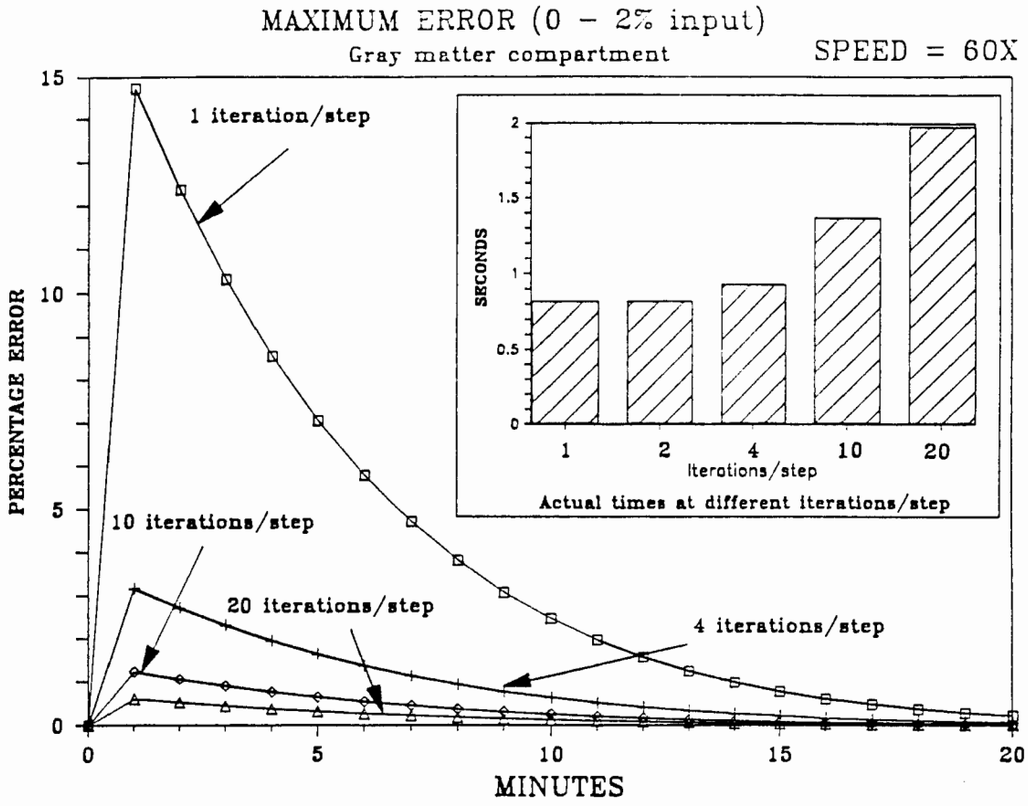


Figure 5.6 Error in a "fast" compartment at a speed of 60 times real time for a 0 to 2% step in input partial pressure. The simulation was repeated for 1, 4, 10, and 20 iterations between graphics updates. The inset bargraphs indicate the actual time in seconds that the computer took to generate 20 minutes of simulated data.

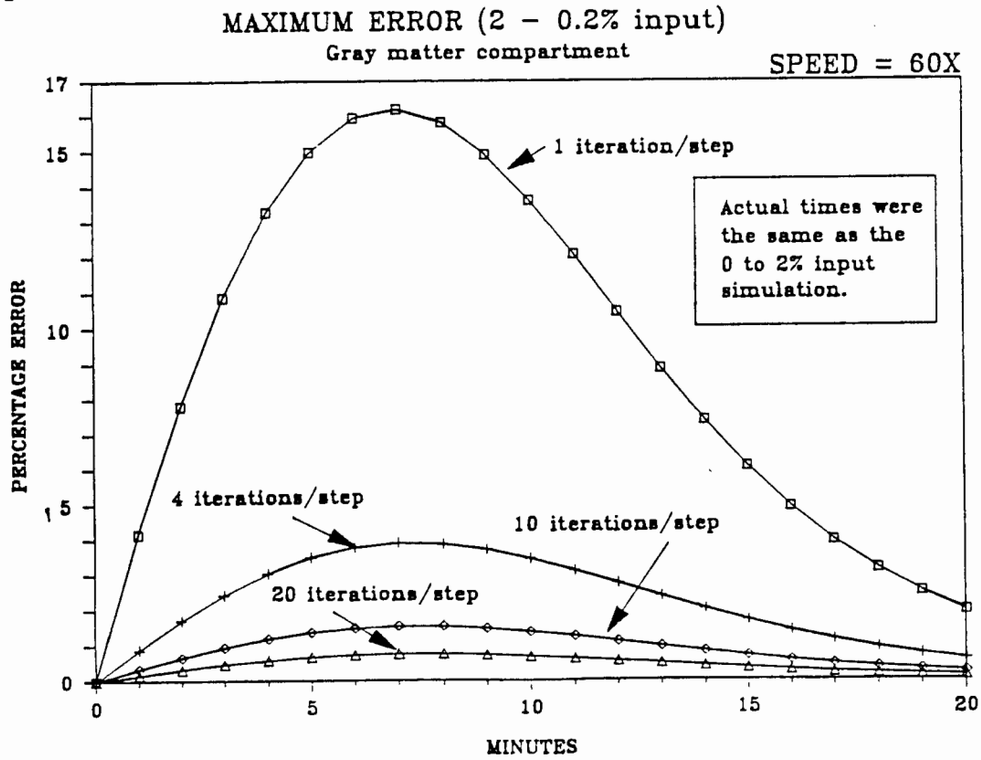


Figure 5.7 Error in a fast compartment at a speed of 60 times real time for a 2 to 0.2 % step in input partial pressure.

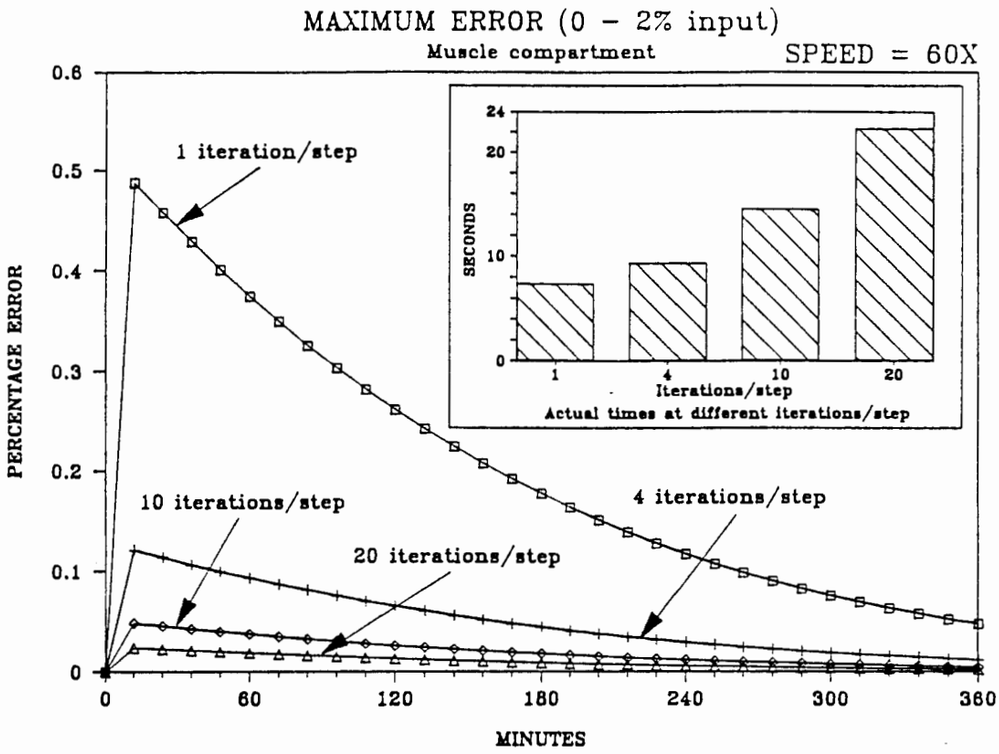


Figure 5.8 Error in a "slow" compartment at a speed of 60 times real time for a 0 to 2% step in input partial pressure. The simulation was repeated for 1, 4, 10, and 20 iterations between graphics updates. The inset bargraphs indicate the actual time in seconds that the computer took to generate 360 minutes of simulated data.

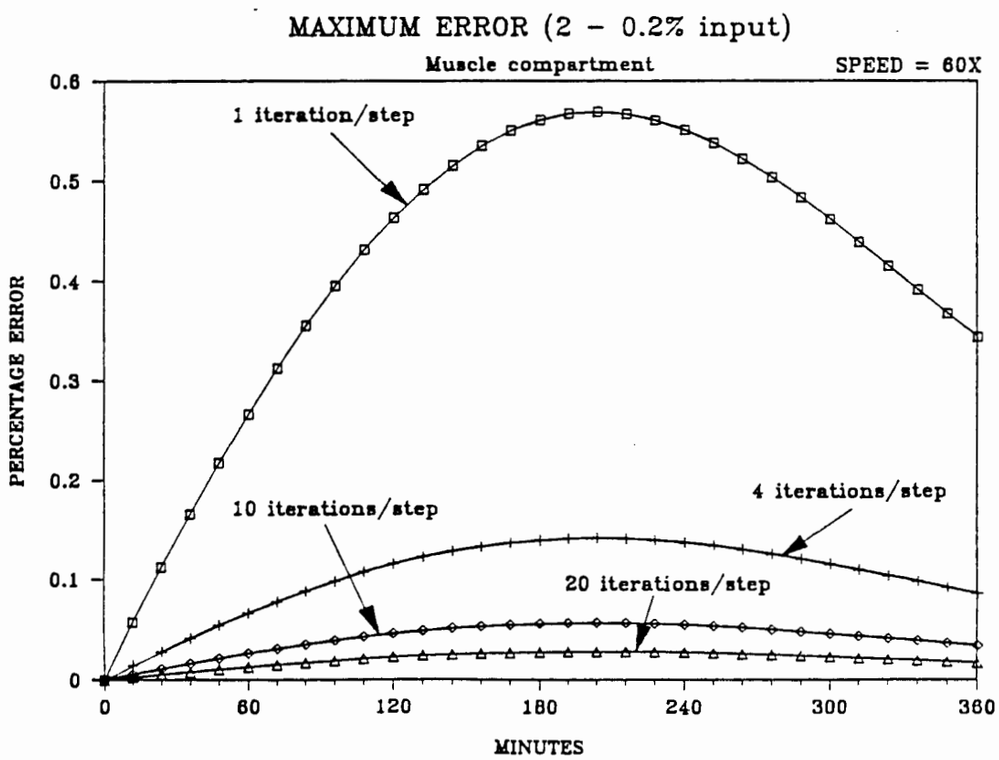


Figure 5.9 Error in a slow compartment at a speed of 60 times real time for a 2 to 0.2 % step in input partial pressure.

5.2 RESULTS FROM THE SIMULATOR USING THE DEMONSTRATION MODEL.

The data generated by the simulator is presented in three sections, each one representing data related to one of the three sub-models which make up the demonstration model. In all cases the model was run at a speed of 60 times real time for a step change in halothane partial pressure introduced into the breathing circuit of 0 to 2%. The breathing rate was maintained at 16 breaths a minute and the tidal volume set to 0.375 litres. The fresh gas inflow rate was maintained at 8 litres/minute for all the simulations except for two of the breathing circuit ones.

5.2.1 THE BREATHING CIRCUIT.

The breathing circuit response is dependent on the flow rate of the fresh gas coming into the circuit compared to the total ventilation rate. When the fresh gas flow is greater than the ventilation rate then the concentration of agent in the inspired air is equal to the concentration of the fresh gas adjusted to include saturation with water vapour. If the fresh flow is less than the ventilation rate then the breathing circuit and inspired concentration will take some time to equilibrate with the fresh gas concentration. Figures 5.10, 5.11, and 5.12 illustrate the response of the breathing circuit to fresh gas flow rates of 8, 4, and 2 litres/minute.

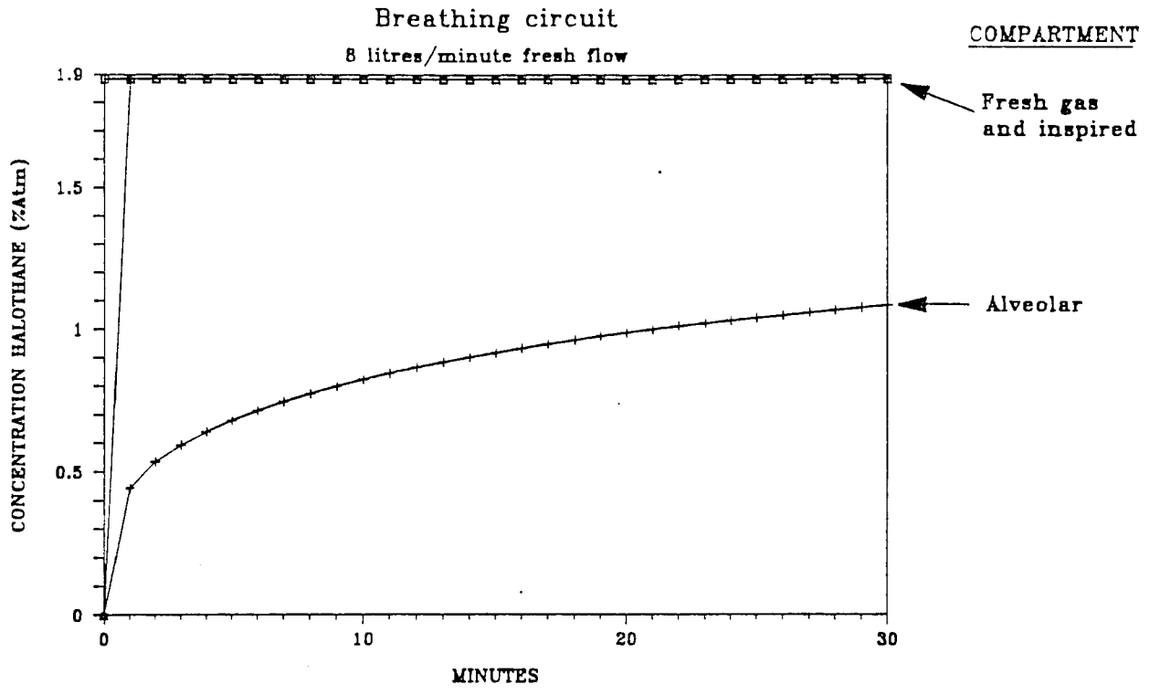


Figure 5.10 Response of the breathing circuit with a fresh gas inflow rate of 8 litres per minute. Fresh halothane partial pressure = 2%; breathing rate = 16 breaths/minute; tidal volume = 0.375 litres.

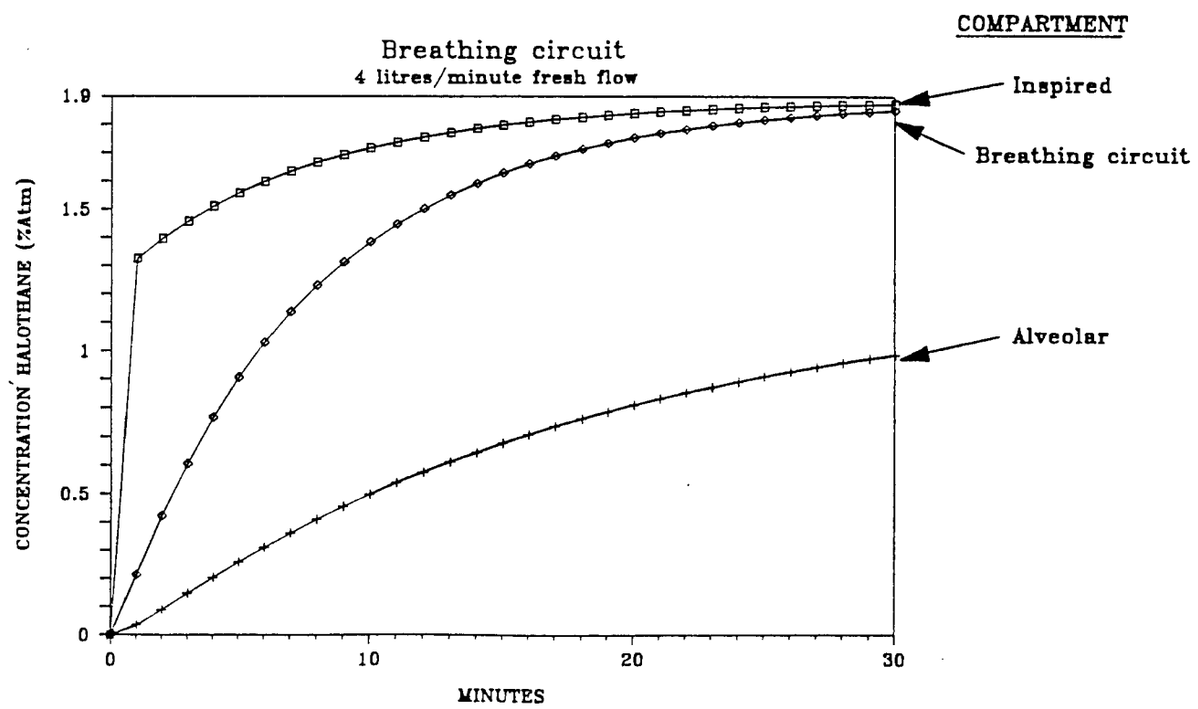


Figure 5.11 Response of the breathing circuit with a fresh gas inflow rate of 4 litres per minute. Control settings are the same as for figure 5.10.

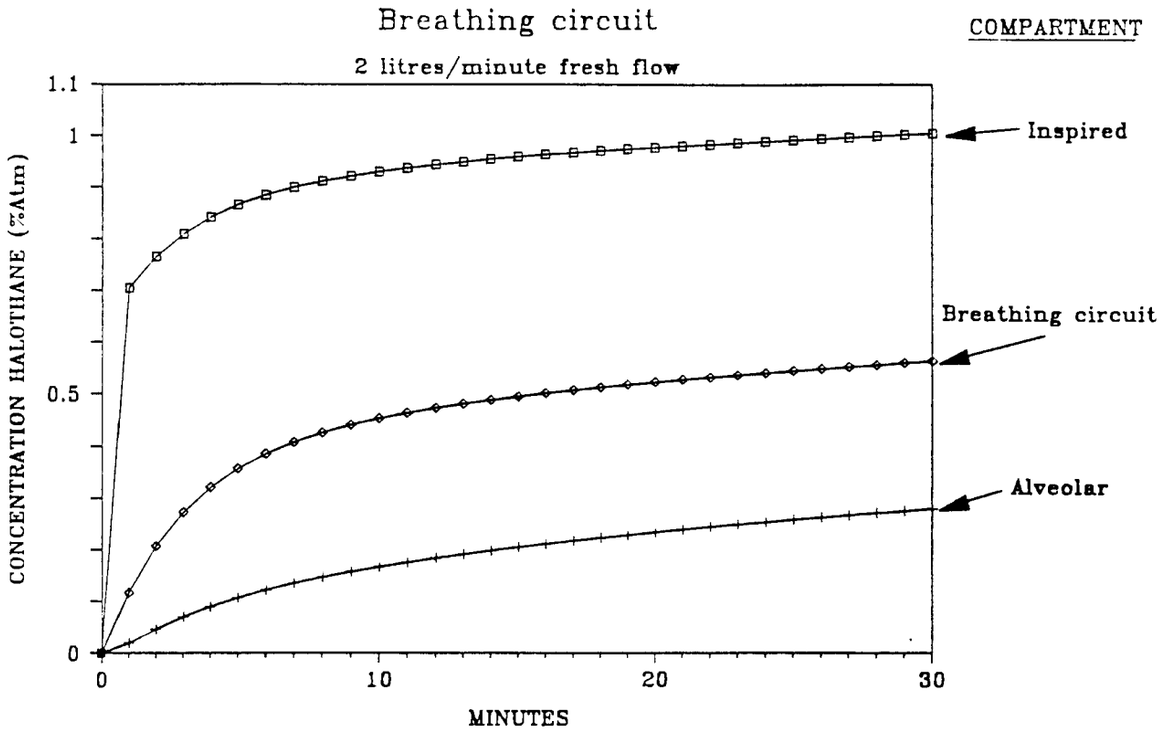


Figure 5.12 Response of the breathing circuit with a fresh gas inflow rate of 2 litres per minute. Control settings are the same as for figure 5.10.

5.2.2 THE UPTAKE AND DISTRIBUTION SUB-MODEL RESULTS.

Graphs of the change in partial pressure of halothane in various representative compartments of the demonstration model are presented in figures 5.13 and 5.14. The first graph shows the changes that occur in six of the thirteen model compartments during the first 12 minutes of a simulation at a speed of 15 times real time. The second graph shows the changes that occur during the first hour of a simulation at a speed of 60 times real time. In both cases the partial pressure of halothane delivered to the breathing circuit was changed from 0 to 2%.

Figures 5.15 and 5.16 show the amount of halothane taken up by various compartments over a time period. The first graph shows the amount taken up by tissues with a small capacity for halothane over 4 hours of simulated time at a speed of 60 times real time. The second graph shows the amount taken up by the muscle compartment (which has a large capacity for halothane) over a period of 12 hours of simulated time. The poorly perfused compartment and the visceral compartment are included for comparison with figure 5.15.

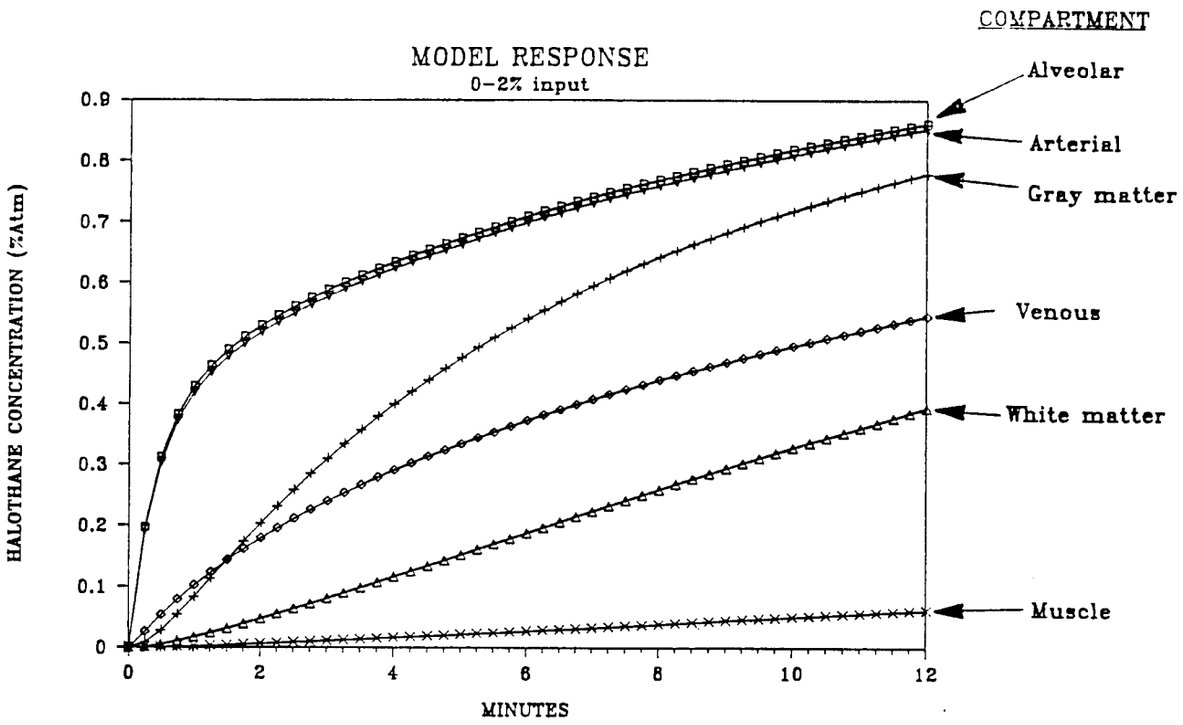


Figure 5.13 Model response to a 0 to 2% step change in halothane partial pressure delivered to the breathing circuit. The graph illustrates the first 12 minutes at a speed of 15 times real time. The difference between the alveolar and arterial concentrations is due to the lung shunt.Fresh gas flow rate = 8 l/minute, Breathing rate = 16 breaths/minute, Tidal volume = 0.375 litres.

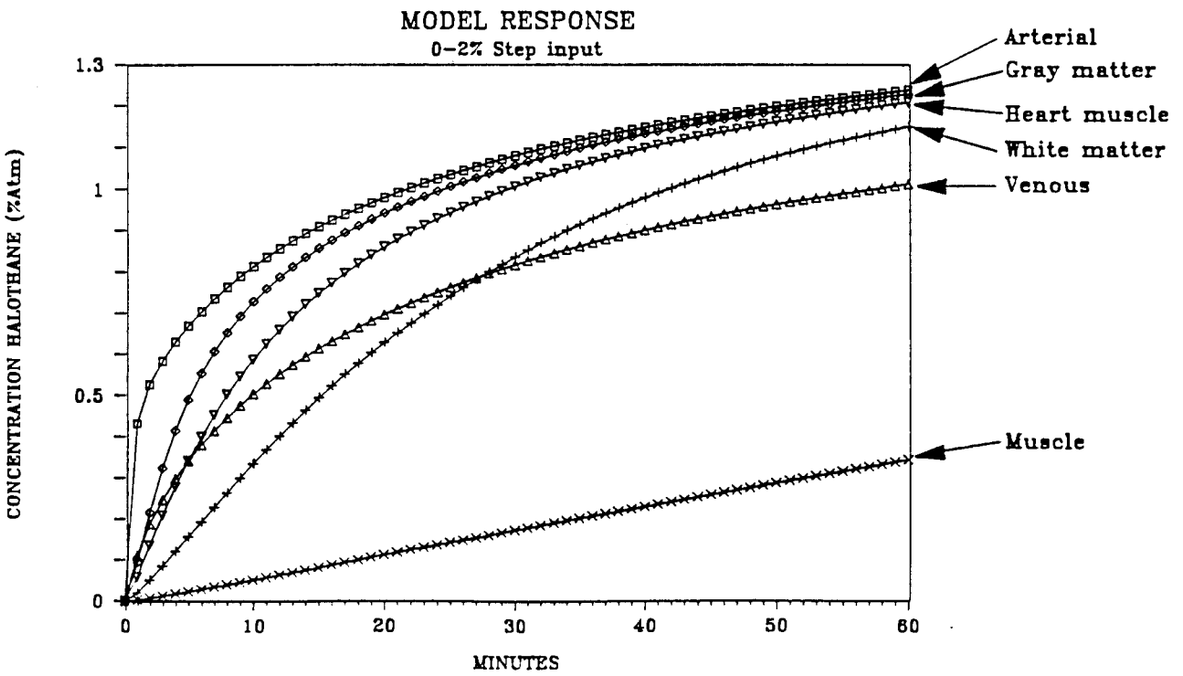


Figure 5.14 Model response to a 0 to 2% step change in halothane partial pressure delivered to the breathing circuit. The graph illustrates the first 60 minutes at a speed of 60 times real time.

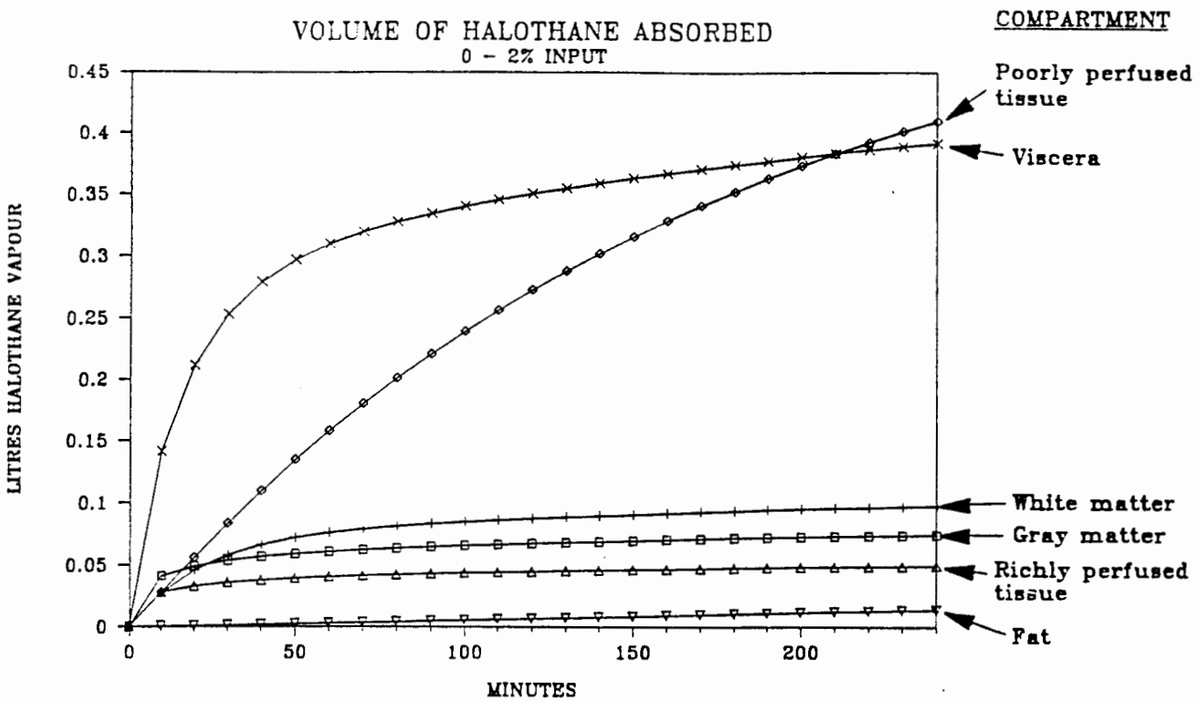


Figure 5.15 The amount of halothane taken up by various model compartments over 4 hours of simulated time. Fresh gas flow rate = 8 l/minute, Breathing rate = 16 breaths/minute, Tidal volume = 0.375 litres.

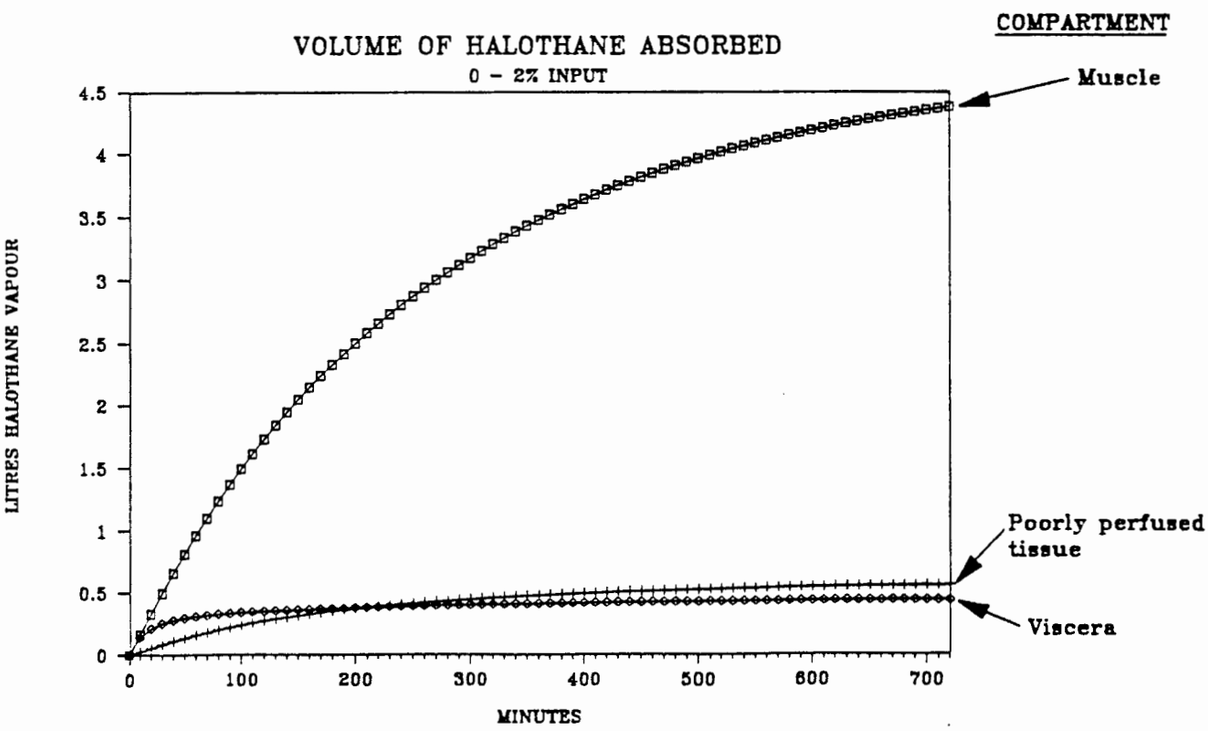


Figure 5.16 The amount of halothane taken up by the muscle compartment over 4 hours of simulated time. Fresh gas flow rate = 8 l/minute, Breathing rate = 16 breaths/minute, Tidal volume = 0.375 litres.

5.2.3 THE CARDIOVASCULAR SUB-MODEL.

The response of the cardiovascular system sub-model is presented in this section. In all cases the partial pressure of halothane was changed from 0 to 2%, the fresh gas flow rate was 8 l/minute, the breathing rate was 16 breaths/minute, and the tidal volume was 0.375 litres. The simulation was performed at 60 times real time.

Figure 5.17 illustrates the change in blood pressure that occurs over 60 minutes of simulated time. Figure 5.18 shows the response of the heart rate, cardiac output, and stroke volume to the anaesthetic gas. Figure 5.19 shows the decrease in peripheral resistance that occurs.

5 of the model compartments have variable conductances. Figures 5.20 and 5.21 show the changes that occur in the conductance and the blood flow rate through these compartments over a 60 minute period.

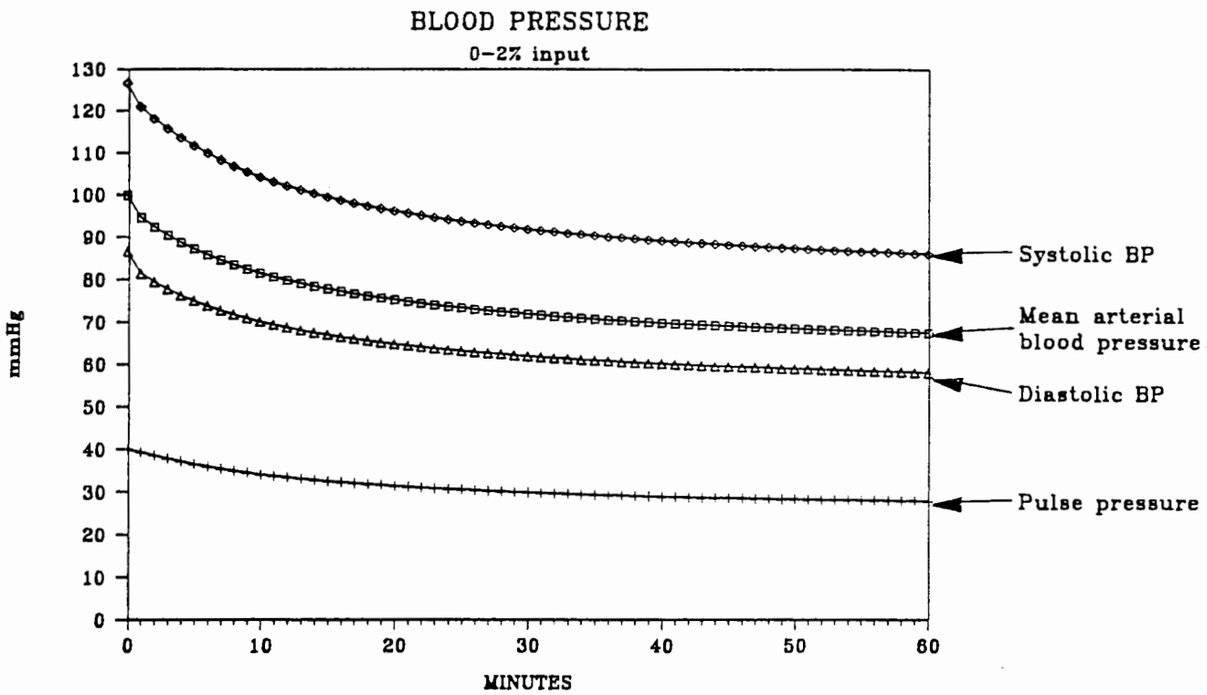


Figure 5.17 Change in blood pressure in response to a 0 to 2% step change in halothane partial pressure introduced into the breathing circuit.

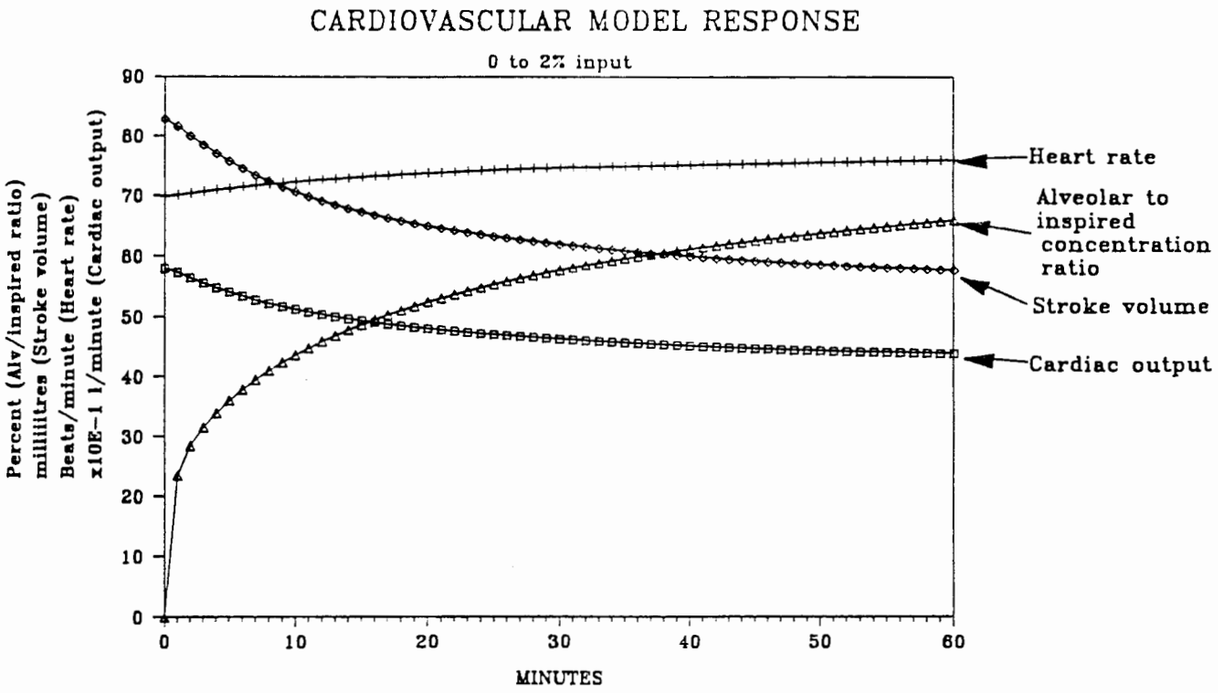


Figure 5.18 Response of the cardiovascular sub-model to a step change in partial pressure delivered to the breathing circuit of 0 to 2% halothane. Note that there are 4 different parameters being plotted against the y axis. The cardiac output data was multiplied by a factor of 10 to allow it to be seen more clearly.

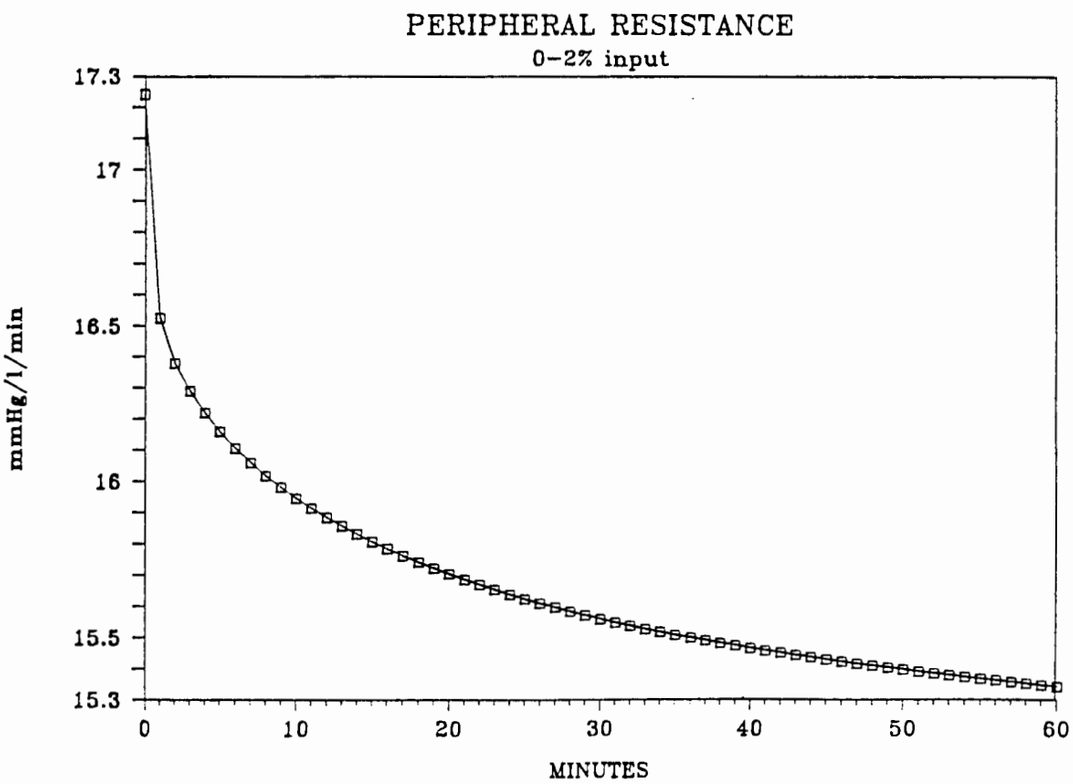


Figure 5.19 Change in peripheral resistance due to step input of 0 to 2% fresh halothane.

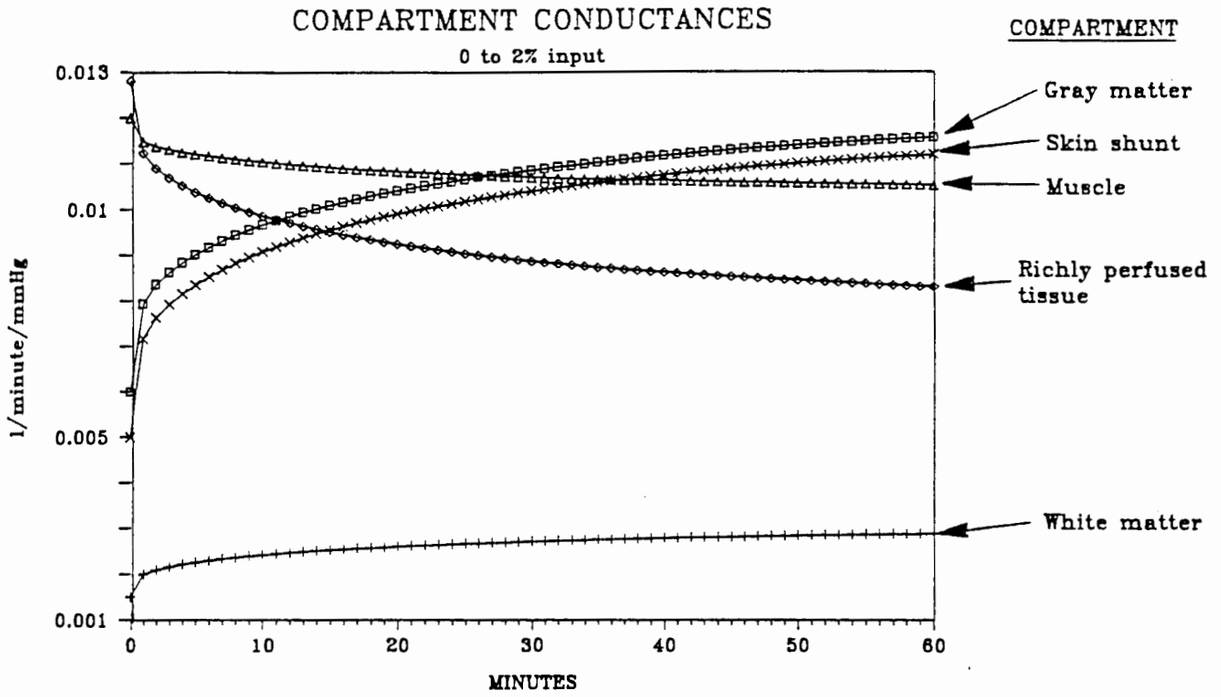


Figure 5.20 Response of compartment conductances to a step change in partial pressure delivered to the breathing circuit of 0 to 2% halothane.

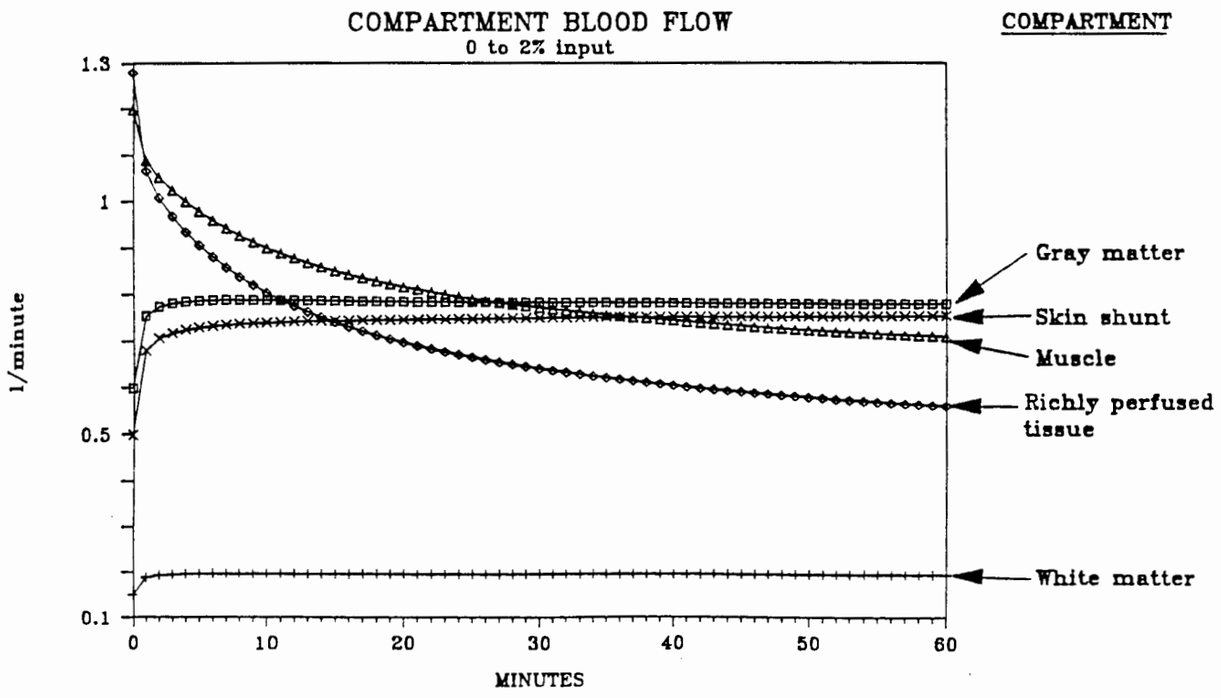


Figure 5.21 Response of blood flows in compartments with variable conductance to a step change of 0 to 2% fresh halothane.

5.3 SIMULATOR RESPONSE TIME.

The simulator response time depends on the simulation speed. It also depends on the number of parameters being graphed. Figure 5.22 shows the actual time in seconds taken for the simulator to calculate 5 minutes worth of data at different speeds. The results are shown for when the graph screen is not active and just bargraphs are being plotted, when the graph screen is active and plotting 6 parameters, and for when the graph screen is active and only plotting 1 parameter.

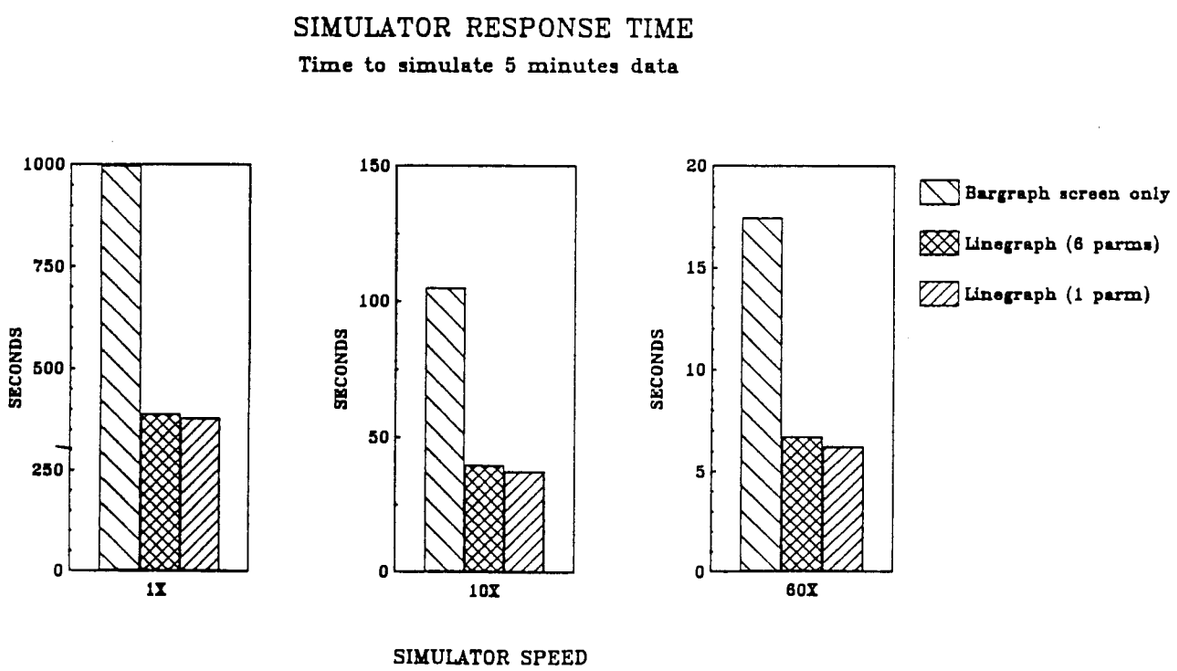


Figure 5.22 Simulator response time. The bargraphs show the actual time in seconds taken for the simulator to simulate 5 minutes of data at various speeds and for different graph screens.

CHAPTER 6

DISCUSSION.

CHAPTER 6

DISCUSSION

A review of the literature revealed that modelling the uptake and distribution of anaesthetic agents has been conducted since the 1950's. Modelling techniques have changed over the years. The first simulations were performed on analogue computers (Mapleson 1963). Digital computers have been used since the 1960's and recently a simulation was performed using a combination of both analogue and digital computers (Fukiu and Smith 1981).

The models used have evolved from simple linear ones to the current complicated non-linear multiple models. The idea of breaking up a model into a number of different compartments consisting of similar tissue types is well established. Simplifying assumptions about these compartments have been described which allow differential equations that describe the uptake and distribution of anaesthetic gases in these compartments to be derived. These have all been tested and improved on by various researchers and have been shown to give fairly realistic results.

6.1 THE MODEL

The demonstration model developed for the simulator consists of a combination of features from various existing models. For convenience and simplicity the described model only considers the anaesthetic agent halothane. The uptake of the respiratory gases oxygen and carbon dioxide and the effect

these gases have on the cardiovascular system are neglected. The effects of other anaesthetic agents such as nitrous oxide are also neglected. Obviously these effects should be included in future versions of the simulator. It was not practical to do so in this case as the calculations required for an additional three uptake and distribution sub-models would have slowed the simulator response time down to an unacceptable level. Future improvements to the model will have to be accompanied by improvements in the computer system to improve the speed of floating point calculations.

6.2 THE SIMULATION TECHNIQUE.

Even if the model was a perfect representation of a human patient it would be impractical so simulate the response of the model to an anaesthetic procedure with perfect accuracy on a digital computer. This is due to the quantization error caused by calculating the change in the model parameters over a series of time steps. The smaller the step, the more accurate the result. However this leads to an increasing number of calculations that have to be performed by the computer. This in turn leads to long periods of waiting for the results to appear on the screen and may cause roundoff errors as well. Thus a trade-off has to be made between accuracy and operating speed.

A program was written to help make this descision. A simulation of the change of concentration in a single compartment was run and the results compared with the exact answer obtained after solving the differential equation describing that compartment. These results are presented in

the previous chapter in figures 5.1 to 5.9. The results confirm that smaller step sizes (ie. more iterations per step) give smaller errors. However it was found that for the same step size, compartments whose concentrations changed very rapidly compared to a slower compartment gave a larger maximum error. Comparing figure 5.6 with figure 5.8 which shows the error generated in the grey matter and the muscle compartments at a speed of 60 times real time, one can see that the slower uptake muscle compartment has an error of less than 0.5% at all step sizes compared to the faster uptake grey matter compartment which has a maximum error of less than 15%.

In all the graphs it can be seen that the error eventually converges to zero as the compartmental concentration equilibrates with the concentration in the arterial blood supply. The faster uptake compartments, such as the grey matter compartment which generates a larger error, equilibrate faster thus cancelling out the effect of this larger error.

From the curves it can be seen that 10 iterations between each graphics update gives an error of less than 2% for both compartments at all speeds up to 60 times real time (the slower the speed, the less the error). This value was chosen for the simulator program. ie, at a speed of 1 times real time, the iteration step size is 1 second/10 which equals 0.1 of a second and at 60 times real time the iteration step size is 1 minute/10 which equals 0.1 of a minute or 6 seconds. The response of the simulator was then timed at different speeds while plotting the data on different graphics screens and the

results presented in figure 2.22. These results indicate that updating the bar-graphs takes more than twice as long as updating the graph window only. There is a slight difference in time taken between updating the graph window when only 1 parameter is being plotted and when 6 parameters are being plotted.

The simulator response time is slower than the simulated time interval when operating at a speed of 1 times real time. ie. the simulator takes almost 1000 seconds to simulate 300 seconds of data when updating the bar-graphs and just less than 400 seconds when plotting the data in the graph window alone. Thus for teaching purposes it would be preferable to run the simulator at 60 times real time while plotting the desired parameters on the graph screen. In this way there will be a minimum of waiting while the simulator runs and trends can be observed at the same time.

6.3 THE RESPONSE OF THE DEMONSTRATION MODEL.

The response of the demonstration model to a simulated change in partial pressure of halothane delivered to the breathing circuit of 0 to 2% is presented in three parts in the previous section, each part representing data generated by each of the sub-models.

6.3.1 THE BREATHING CIRCUIT RESPONSE.

The response of the breathing circuit, shown in figures 2.10, is dependent on the flow rate of fresh gas entering the breathing circuit and on the breathing rate and tidal volume

of the patient. A high fresh gas flow rate greater than the total ventilation rate prevents rebreathing of expired gas. Expired gas and a percentage of the fresh gas is vented through the pop-of valve. Thus the concentration of gas in the breathing circuit is equal to the fresh gas concentration as shown in figure 5.10.

The default values of breathing rate and tidal volume (16 breaths/minute and 0.375 litres) give a total ventilation rate of 6 litres/minute and an alveolar ventilation rate of 3.6 litres/minute. When the fresh gas flow gets less than 6 l/minute but greater than 3.6 l/minute the breathing circuit and inspired concentrations of halothane behave as shown in figure 5.11. The concentration in the breathing circuit (or the CO_2 absorber) consists of expired dead space gas which has the same concentration as the inspired gas(the alveolar gas is lost through the pop-of valve). The inspired gas consists of a mixture of this breathing circuit gas and the fresh gas flowing into the circuit.

When the fresh gas flow rate falls below the alveolar ventilation rate of 3.6 l/minute, the breathing circuit gas is made up of a mixture of expired dead space gas and alveolar gas. The inspired gas concentration (illustrated in figure 2.12) does not equilibrate with the fresh gas as the delivery of fresh gas to the breathing circuit is less than the removal of gas by the lungs.

6.3.2 THE UPTAKE AND DISTRIBUTION SUB-MODEL RESPONSE.

A representative group of seven compartments were chosen to illustrate the change in partial pressure that occurs over a time interval in the model. Figures 5.13 and 5.14 show the response of these compartments over a 12 minute and a 60 minute interval. The fast compartments such as the arterial and the grey matter compartment approach the inspired partial pressure (2%) fairly rapidly. The slower compartments such as muscle tissue (which has a large capacity for halothane) take much longer to approach equilibrium. These results agree very closely with those published by Smith et al (1972) which indicates that the simulator is working correctly.

A feature that was included in the simulator is to enable the operator to graph the actual volume (or amount) of halothane vapour taken up by any compartment over a certain time period. Thus one can observe a compartment such as the viscera approach saturation. The capacity of the viscera for halothane when in equilibrium at a partial pressure of 2% is approximately 0.5 litres of vapour. The way in which this compartment and various others with different capacities approach their maximum capacity with time is shown in figures 5.15 and 5.16. From these graphs it can be seen that compartments with large capacities for halothane take longer to reach saturation.

6.3.3 THE CARDIOVASCULAR SUB-MODEL RESPONSE.

Certain parameters of the cardiovascular system sub-model are affected by the partial pressure of halothane in specific compartments. The cardiac output and the stroke volume are both linearly depressed with rising halothane concentration in the heart muscle. This effect is illustrated in figure 5.18. The heart rate, which is the cardiac output divided by the stroke volume rises slightly with an increase in heart muscle partial pressure.

The total peripheral resistance decreases with increasing arterial partial pressure. Five of the model compartments have variable conductances. The conductance of the grey matter, white matter and skin shunt compartments increase and that of the muscle and richly perfused compartments decrease. This has the effect of increasing and decreasing the blood flow through the compartments respectively. This is illustrated in figures 5.20 and 5.21.

The decrease in cardiac output and in peripheral resistance (the inverse of peripheral conductance) leads to a decrease in mean arterial blood pressure. Pulse pressure is assumed to be roughly proportional to stroke volume and is used to calculate systolic and diastolic blood pressures as illustrated in figure 5.17.

6.4 THE COMPUTER HARDWARE

The simulator program, running on an IBM PC XT with a clock frequency of 10Mhz, was found to be slightly slow when performing real time simulations (as shown in figure 5.22). However the response of the simulator was found to be satisfactory at simulation speeds faster than real time (it takes less than 18 seconds to simulate 300 seconds of data at a speed of 60 times real time).

The program and all the accessory files fit onto a single 320 kilobyte 5 1/4 inch floppy disk and is also capable of being run from a hard drive. The program needs at least 512 kilobytes of memory to run (most of the memory is needed by the graphics routines).

The monochrome graphics card has a high resolution that allows a greater amount of data to be displayed on the screen than if a color graphics adapter was used. Thus this version of the simulator will only run on a computer with a monochrome graphics card fitted.

CHAPTER 7

CONCLUSIONS AND PROPOSALS FOR FUTURE STUDIES

CHAPTER 7

CONCLUSIONS AND PROPOSALS FOR FUTURE STUDY.

7.1 CONCLUSIONS.

This study has shown that :

1) Although the uptake and distribution of anaesthetic agents has been modelled extensively since 1951, these models have generally not been used to supplement the training of student anaesthetists.

2) It was possible to develop a generalised n compartmental non-linear multiple model of the uptake and distribution of halothane by combining features of various existing models. Raw data for a demonstration model was obtained from results published by the authors of other models.

3) The maximum error due to Euler integration on a digital computer could be kept less than 2% if the time step over which each calculation took place was kept small enough. A test program was developed to determine the maximum usable step size which was found to be 0.1 minutes or 6 seconds.

4) When the model is iterated over a small time step of 0.1 seconds the simulator response time is slow. ie the program runs for 1000 seconds to simulate 300 seconds of data. However the response of the simulator is satisfactory when larger iteration step sizes are used.

5) It is possible to implement a large mathamatical model on an IBM personal computer and run real time and faster than real time simulations on it. The demonstration model developed for the simulator was found to be approaching the limit that would allow real time simulations to be performed on a simple IBM PC XT operating with a clock frequency of 10Mhz. Implementation of the simulator on an IBM AT with a math coprocessor installed would allow the size of the model to be increased.

6) The monochrome graphics display had a fine enough resolution to display a large amount of information at any one time. However the lack of colour was found to be confusing and the variable sized text font used in graphics mode is not very legible.

7) At this stage the simulator has certain limitations. The lack of oxygen and carbon dioxide sub-models as well as the absence of sub-models describing the effects of other anaesthetic agents such as nitrous oxide need to be considered in an expanded model.

This initial version of the simulator however should provide a valuable teaching aid to anaesthetists and others wishing to become acquainted with the pharmacokinetics of halothane uptake by a human subject.

7.2 PROPOSALS FOR FUTURE STUDY.

A number of improvements to the model and computer hardware could enhance the teaching capabilities of the simulator.

1) An expanded model is required. Sub-models of the uptake and distribution of oxygen and carbon dioxide and their effect on the cardiovascular and respiratory systems is an obvious addition. It may be possible to adapt an existing model such as "MacPuff" to this purpose. The uptake and distribution of other commonly used anaesthetic agents should be included. Various other breathing circuit arrangements should be supported. Some other improvements could include the support of different "patients", male and female, with different ages, weights and adjustable cardiac output.

2) A faster computer will be needed if the model is to be improved on. The current IBM PC XT will not be able to perform the required calculations fast enough. However the addition of a maths coprocessor to an XT machine would almost double the operating speed of the current simulator.

3) Better graphics are required. An enhanced graphics adapter will give 16 colours on a screen with almost the same resolution as that of the monochrome screen. However to be compatible with most other IBM's the program should also support the other commonly used graphics cards.

4) Different integration techniques may be tried with a new model. The improved Euler method requires more than twice the

amount of calculations than the simple Euler method but gives a much smaller error.

5) A tutorial could be included in the program to enhance the training potential of the simulator.

The implementation of these improvements would make the simulator more useful as an aid towards helping students to understand the pharmacokinetics of the uptake and distribution of anaesthetic agents and there effects on the human cardiovascular system.

Should the above improvements be implemented then the next step would be to interface hardware components to the computer system such as a "patient" manikin, physiological monitors, and an anaesthesia machine. Thus future students may be able to obtain practical as well as theoretical skills in anaesthesia with the knowledge that they may practice without the possibility of harming or inconveniencing a patient.

REFERENCES

REFERENCES.

ABRAHAMSON S, DENSON JS, WOLF RM.
1969.
Effectiveness of a simulator in training anaesthesiology residents.
Journal of Medical Education, 44:515-519.

ASHMAN MN, BLESSER WB, EPSTEIN RM.
1970.
A non-linear model for the uptake and distribution of halothane in man.
Anesthesiology, 33(4):419-429.

ATKINSON RS, RUSHMAN GB, LEE JA.
1987.
A synopsis of anaesthesia.
10th edition, Bristol Wright.

BELLVILLE JW, HARA HH.
1966.
Use of analogue computers in anesthetic research.
Anesthesiology, 27(1):70-83.

BENEKEN JEW.
1972.
Some computer models in cardiovascular research.
Ch 6, pp 173-223 In: BERGEL DN ed, Cardiovascular fluid dynamics. Vol 1, Academic Press, London.

BENEKEN JEW, RIDEOUT VC.
1968.
The use of multiple models in cardiovascular systems studies: Transport and perturbation methods.
IEEE Transactions on Biomedical Engineering, 15:281-289.

BRISTOW JD, PRYS-ROBERTS C, FISHER A, PICKERING TG.
1969.
Effects of anesthesia on baroreflex control of heart rate in man.
Anesthesiology, 31(5):422-428.

CLIFTON BS, HOTTEN WIT.
1963.
Deaths associated with anaesthesia.
British Journal of Anaesthesia, 35:250-259.

COOPER JB, NEWBOWER RS, LONG CD, MCPEEK B.
1978.
Preventable anesthesia mishaps : A study of human factors.
Anesthesiology, 49:399-406.

COWLES AL, BORGSTEDT HH, GILLIES AJ.
1971.
Tissue weights and rates of blood flow in man for the prediction of anesthetic uptake and distribution.
Anesthesiology, 35(5):523-526.

COWLES AL, BORGSTEDT HH, GILLIES AJ.
1973.
A simplified digital method for predicting anaesthetic uptake and distribution.
Computers in Biology and Medicine, 3:385-395.

DAVISON MHA.
1968.
The Evolution of Anaesthesia.
John Sherratt and Son, Altringham.

DENSON JS, ABRAHAMSON S.
1969.
A computer controlled patient simulator.
Journal of the American Medical Association, 208(3):504-508.

DICKINSON CJ.
1972.
A digital computer model to teach and study gas transport and exchange between lungs, blood, and tissues ("McPuff").
Journal of Physiology, 224:7-9.

DICKINSON CJ.
1977.
A computer model of human respiration for teaching, research and clinical use.
MTP Press, Lancaster.

DRIPPS RD, LAMONT A, ECKENHOFF JE.
1961.
The role of anaesthesia in surgical mortality.
Journal of the American Medical Association, 178:261-266.

EGER EI II.
1963a.
A mathamatical model of uptake and distribution.
pp. 72-87 In : PAPPER EM AND KITZ RJ, eds. Uptake and distribution of anaesthetic agents, McGraw-Hill, New York.

EGER EI II.
1963b.
Effect of inspired anesthetic concentration on the rate of rise of alveolar concentration.
Anesthesiology, 24(2):153-157.

EGER EI II.
1974.
Anaesthetic uptake and action.
Williams and Wilkins Company, Baltimore.

EGER EI II, GUADAGNI NP.
1963.
Halothane uptake in man at constant alveolar concentration.
Anesthesiology, 24(3):299-304.

EGER EI II, HALSEY MJ.
1970.
Dimensions of anesthetic uptake.
(Editorial) Anesthesiology, 33(4):381-382.

EGER EI II, SAIDMAN LJ, BRANDSTATOR B.
1965.
Minimum alveolar anesthetic concentration: a standard of anesthetic potency.
Anesthesiology, 26:756-763.

EGER EI II, SMITH NT, STOELTING RK.
1968.
The cardiovascular effects of various alveolar halothane concentrations in man.
Anesthesiology, 29:185-187.

EGER EI II, SMITH NT, STOELTING RK.
1970.
Cardiovascular effects of halothane in man.
Anesthesiology, 32:396-409.

EPSTEIN RM, RACKOW H, SALANITRE E, WOLF GL.
1964.
Influence of the concentration effect on the uptake of anesthetic mixtures: The second gas effect.
Anesthesiology, 25(3):364-371.

FELL D.
1985.
The practical conduct of anaesthesia.
Ch 17, pp 245-256 In: SMITH G, AITKENHEAD AR, eds, Textbook of Anaesthesia, Churchill Livingstone, Edinburgh.

FUKIU Y, SMITH NT.
1981a.
Interactions among ventilation, the circulation, and the uptake and distribution of halothane - Use of a hybrid computer multiple model: I. The basic model.
Anesthesiology, 54:107-118.

FUKIU Y, SMITH NT.
1981b.
Interactions among ventilation, the circulation, and the uptake and distribution of halothane - Use of a hybrid computer multiple model: II. Spontaneous vs controlled ventilation, and the effect of CO2.
Anesthesiology, 54:119-124.

HARRISON GG.
1968.
Anaesthetic contributory death - its incidence and causes. Part 1 : Incidence; Part 2: Causes.
South African Medical Journal, 42:514.

HEFFERNAN PB, GIBBS JM, MCKINNON AE.
1982a.
Teaching the uptake and distribution of halothane. A computer simulation program.
Anaesthesia, 37(1):9-17.

HEFFERNAN PB, GIBBS JM, MCKINNON AE.
1982b.
Evaluation of a computer simulation program for teaching halothane uptake and distribution.
Anaesthesia 37(1):43-46.

HOLADAY DA, SMITH FR.
1981.
Clinical characteristics and biotransformations of sevoflurane in healthy human volunteers.
Anesthesiology 54: 100 - 106.

HON D.
1982.
Interactive training in cardiopulmonary resuscitation.
Byte, 7(6):108-138.

HOWELLS TH, EMERY FM, TWENTYMAN JEC.
1973.
Endotracheal intubation training using a simulator.
British Journal of Anaesthesia, 45:400-402.

JACOBSON B, WEBSTER JG.
1977.
Medicine and clinical engineering.
Prentice-Hall, New Jersey.

KETY SS.
1950.
The physiological and physical factors governing the uptake of anesthetic gases by the body.
Anesthesiology, 11(5):517-526.

KETY SS.
1951.
The theory and applications of the exchange of inert gas at the lungs and tissues.
Pharmacology Review, 3:1-41.

LOWE JH, ERNST EA.
1981.
The quantitative practice of anesthesia: Use of a closed circuit.
Williams and Wilkins, Baltimore, London.

MAPLESON WW.
1963.
An electric analogue for the uptake and exchange of inert agents and other agents.
Journal of Applied Physiology, 18:197-205.

MAPLESON WW.
1973.
Circulation time models of the uptake of inhaled anaesthetics and data for quantifying them.
British Journal of Anaesthesia, 45:319-333.

MUNSON ES, BOWERS DL.
1967.
Effects of hyperventilation on the rate of cerebral anaesthetic equilibration. Calculations using a mathematical model.
Anesthesiology, 28:377-381.

MUNSON ES, EGER EI II, BOWERS DL.
1968.
The effects of changes in cardiac output and distribution on the rate of cerebral anesthetic equilibration.
Anesthesiology, 29(3):533-537.

MUNSON ES, EGER EI II, BOWERS DL.
1973.
Effects of anesthetic-depressed ventilation and cardiac output on anesthetic uptake: A computer nonlinear model.
Anesthesiology, 38(3):251-259.

NUNN JF.
1977.
Applied respiratory physiology.
2nd Edition, Butterworths, London.

PHILLIP JH.
1984.
GAS MAN. Understanding anesthesia uptake and distribution.
Addison-Wesley, California.

SAKURAI Y, KIKUCHI M, TANISHITA K, HARADA M, AKIYAMA T.
1980.
Patient robot system for resuscitation training.
pp. 1224-1228 In: LINDBERG DAB, KAIHARA S, eds. Medinfo 80, North Holland Publishing company, Amsterdam.

SECHER O.
1982.
Inhalation anaesthetics.
Acta Anaesthesia Scandinavica, 26:269-278.

SMITH NT, ZWART A, BENEKEN JEW.
1972.
Interaction between the circulatory effects and the uptake and distribution of halothane : Use of a multiple model.
Anesthesiology, 37(1):45-58.

STEWARD A, ALLOT PR, COWLES AL, MAPLESON WW.
1973.
Solubility coefficients for inhaled anaesthetics for water, oil, and biological media.
British Journal of Anaesthesia, 45:282-293.

STOELTING RK, MILLER RD.
1984.
Basics of anesthesia.
Churchill Livingstone, New York.

TANNER G.
1982.
Pharmacokinetics of inhalation anaesthetics: A three compartmental linear model.
Anesthesia and Analgesia, 61(7):587-594.

TOMPSON KD, SPAIN JA, CALKINS JM, THOMPSON BM, VANDERLEI PM.
1986.
Gas uptake simulator for medical practitioners/educators.
pp. 259 - 261 IN : SALAMON R, BLUM B, JORGENSON M, eds.
MEDINFO 86, Elsevier Science Publishers B.V. (North Holland).

TOUNG TJK, DONHAM RJ, ROGERS MC.
1984.
The stress of giving anesthesia on the electrocardiogram (ECG) of anesthesiologists (Abstract).
Anesthesiology, 61(3A):A465.

VICKERS MD, SCHNIEDEN H, WOOD-SMITH FG.
1984.
Drugs in anaesthetic practice.
Butterworths, London, 6th edition.

WYANT GM.
1978.
Mechanical misadventures in anesthesia.
University of Toronto press, Toronto.

ZWART A, SMITH NT, BENEKEN JEW.
1972.
Multiple model approach to uptake and distribution of halothane: The use of an analogue computer.
Computers and Biomedical Research, 5:228-238.

APPENDIX A

TABLES OF DATA USED IN THE MODEL EQUATIONS.

APPENDIX A

TABLES OF DATA USED IN THE MODEL EQUATIONS

TABLE A.1

MISCELANEOUS DATA

Parameter	Value	
Functional residual capacity (FRC)	2.50	L
Initial breathing rate (BR0)	16.0	B/Min
Initial cardiac output (Q0)	5.80	L/Min
Initial mean arterial blood pressure (MABP)	100	mmHg
Initial pulmonary tidal volume ($V_{tidal}0$)	0.375	L
Lung shunt (shunt)	3	%Q0
Physiological dead space (V_{dead})	0.15	L
Volume of breathing circuit (V_{bc})	15	L

TABLE A.2

COEFFICIENTS, BLOOD VOLUMES AND TISSUE VOLUMES FOR A 5 KG MAN

Model compartment	Tissue volume (L)	Blood Volume (L)	Partition coefficient
Arterial blood	-	0.960	2.3
Fat	12.2	0.160	138
Cerebral grey mater	0.75	0.371	5.4
Myocardium	0.30	0.148	8.1
Splanchnic bed	3.90	0.976	6.0
Lung	0.60	0.372	5.3
Muscle	33.0	0.410	8.1
Poorly perfused tissue	6.20	0.117	5.3
Richly perfused tissue	0.34	0.876	3.7
Skin shunt	-	0.400	-
Venous blood	-	0.560	-
Cerebral white mater	0.75	0.100	8.3

After Zwart et al (1972).

TABLE A.3

BLOOD FLOWS : AWAKE AND AT 2% END TIDAL HALOTHANE

Model compartment	Blood flow Awake (L/Minute)	Blood flow 2% Halothane (L/Minute)	Percent change (L/Minute)
Arterial blood	5.800	3.480	-40
Fat	0.200	0.100	-50
Cerebral grey mater	0.600	0.750	+25
Myocardium	0.250	0.125	-50
Splanchnic bed	1.430	0.715	-50
Lung	5.626	3.376	-40
Muscle	1.200	0.480	-60
Poorly perfused tissue	0.190	0.095	-50
Richly perfused tissue	1.280	0.277	-78
Skin shunt	0.500	0.750	+50
Venous blood	5.800	3.480	-40
Cerebral white mater	0.150	0.188	+25

After Zwart et al (1972).

TABLE A.4

INITIAL CONDUCTANCES AND CONDUCTANCE EQUATION CONSTANTS

Model compartment	Initial conductance (Gn0 L/Min/mmHg)	Constant bn for Gn = Gn0(1 + bn*Pn)
Fat	0.0020	0
Cerebral grey mater	0.0060	0.75
Myocardial	0.0025	0
Splanchnic bed	0.0143	0
Muscle	0.0120	-0.10
Poorly perfused tissue	0.0019	0
Richly perfused tissue	0.0128	-0.2836
Skin shunt	0.0050	1.00
Cerebral white mater	0.0015	0.7533

The above values were calculated using the data from table A.3 and the equations below. Mean arterial blood pressure at 2% end tidal halothane is assumed to be half of the initial MABP (Smith et al 1972).

$$G_n0 = \frac{Q_n0}{MABP0} \qquad (A.1)$$

and

$$b_n = \left(\frac{Q_n(2)}{MABP \bullet G_n0} - 0.5\right) \qquad (A.2)$$

TABLE A.5

MISCELANEOUS CONSTANTS

Parameter	Value	Units
a	-0.2	$\%Atm^{-1}$
b	-0.24	$\%Atm^{-1}$

These constants are used in the following equations relating cardiac output and stroke volume to halothane concentration in the heart muscle.

$$Q = Q0(1 + a \cdot P_h) \qquad (A.3)$$

$$SV = SV0(1 + b.P_h) \qquad (A.4)$$

APPENDIX B

MODEL PARAMETER ABBREVEATIONS, UNITS, AND SUBSCRIPTS.

APPENDIX B.

MODEL PARAMETER ABBREVIATIONS, UNITS, AND SUBSCRIPTS.

TABLE B.1

SUBSCRIPTS USED TO DISTINGUISH BETWEEN MODEL COMPARTMENTS

Subscript	Model compartment
a	Arterial blood compartment
nb	Blood of compartment n
bc	Breathing circuit
f	Fat compartment
g	Cerebral grey matter
h	Myocardium
i	Splanchnic bed
l	Lung compartment
m	Muscle compartment
p	Poorly diffused tissues
r	Richly diffused tissues
s	Lung shunt
sk	Skin shunt
nt	Tissue of compartment n
v	Venous blood compartment
w	Cerebral white matter

TABLE B.2

MODEL PARAMETER ABBREVIATIONS AND UNITS.

Parameter abbreviation	Parameter description	Units
a	Constant for cardiac output equation.	%Atm_1
A_n	Amount of halothane vapour in compartment n	L
BR	Breathing rate	Breaths/Min
BR0	Initial breathing rate	Breaths/Min
Cap_n	Capacity of compartment n for halothane at 1 atmosphere.	L/100L
DiasP	Diastolic blood pressure	mmHg
dP_ndt	Rate of change of halothane partial pressure in the nth compartment.	
fa_fi	Alveolar to inspired concentration difference	Ratio
FRC	Functional residual capacity	L
G_n	Blood conductance of the nth compartment.	L/Min/mmHg
G_n0	Initial conductance of the nth compartment.	L/Min/mmHg
G_{sum}	The sum of conductances of the nine parallel compartments	L/Min/mmHg
G_{sum}	The initial sum of conductances	L/Min/mmHg
HR	Heart rate	B/Min
HR0	Initial heart rate	B/Min
K_n	Constant for uptake equation of the nth compartment.	
MABP	Mean Arterial blood pressure	mmHg
MABP0	Initial arterial pressure	mmHg
Pamb	Atmospheric pressure	mmHg
Ph20	Water vapour pressure 37½C	mmHg
P_{ins}	Inspired halothane partial pressure	% Atm

Pfr	Halothane partial pressure of fresh gas entering breathing circuit.	%Atm
P_n	Halothane partial pressure in the nth compartment	% Atm
P_{uls}	Pulse blood pressure	mmHg
Q	Cardiac output	L/Min
Q0	Initial cardiac output	L/Min
Q_{alv}	Alveolar minute volume	L/Min
$Q_{alv}0$	Initial alveolar minute volume	L/min
Q_{dead}	Dead space ventilation	L/Min
Q_e	Ventilation minute volume	L/Min
Q_{fr}	Fresh gas flow	L/Min
Q_n	Blood flow through the nth compartment.	L/Min
R_{sum}	Peripheral resistance	mmHg/L/Min
shunt	Lung shunt as a fraction of cardiac output.	
SV	Heart stroke volume	L
SV0	Initial heart stroke volume	L
SystP	Systolic blood pressure	mmHg
V_{bc}	Breathing circuit volume	L
V_{dead}	Pulmonary dead space volume	L
V_{tidal}	Tidal volume	L
$V_{tidal}0$	Initial tidal volume	L
V_{nt}	Volume of tissue in the nth compartment.	L
V_{nb}	Volume of blood in the nth compartment.	L
λ_b	Solubility of halothane in blood.	
λ_n	Solubility of halothane in nth compartment.	

APPENDIX C

EULER INTEGRATION

APPENDIX C

EULER INTEGRATION

Consider the function below :

$$y(t) = y(0) + \int_{t_0}^{t} F(u,y(u))\ du \qquad (C.1)$$

This function is shown in figure C.1. The t axis is divided into a number of small steps, numbered $t_0..t_m$, of size h where $h = t_{n+1} - t_n$ $(0 <= n <= m)$. The corresponding values of the function on the y axis are $y_0...y_m$.

To find an approximation y_1 for $y(t_1)$ for the initial value problem :

$$\frac{dy}{dt} = y'(t) = F(t,y) \quad , \quad y(t_0) = y_0 \qquad (C.2)$$

replace

$$y'(t_0) = \lim_{h->0} \frac{y(t_0 + h) - y(t_0)}{h} \qquad (C.3)$$

with the difference quotient :

$$\frac{y(t_1) - y_0}{t_1 - t_0} \approx y'(t_0) = F(t_0,y_0) \qquad (C.4)$$

to obtain :

$$y(t_1) \approx y_0 + h.F(t_0,y_0) \qquad (C.5)$$

Take the linear approximation $y_0 + h.F(t_0,y_0)$ as an approximation of y_1 to $y(t_1)$ and write :

$$y_1 = y_0 + h.F(t_0,y_0) \qquad (C.6)$$

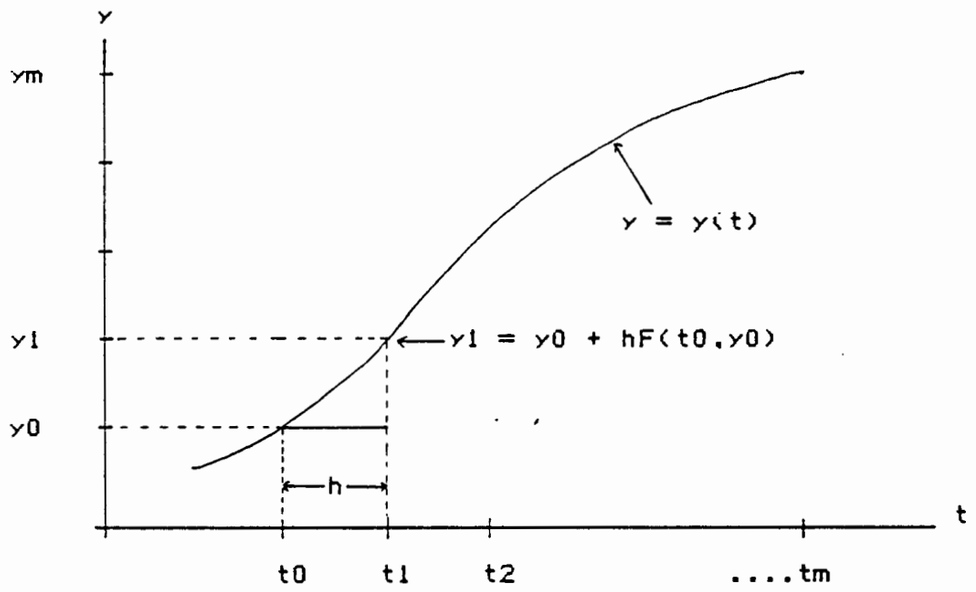


Figure C.1 Euler integration.

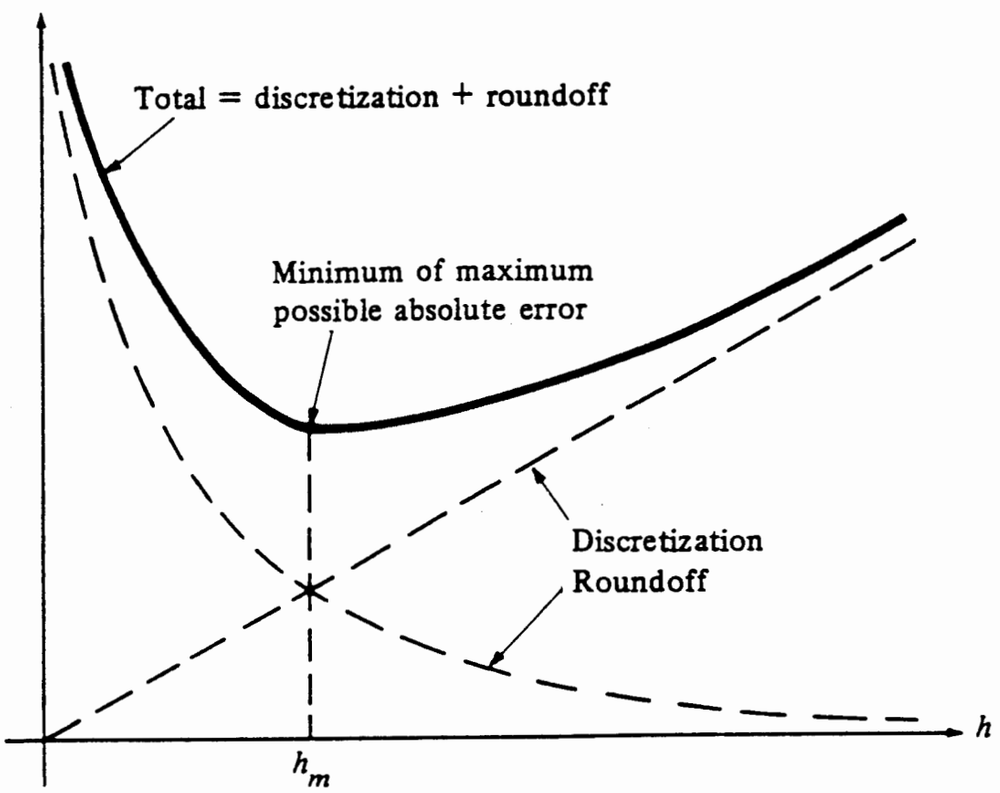


Figure C.2 Errors due to euler integration. The height of the curve equals the maximum possible numerical error.

3

Figure C.2 shows that one is approximating the height of the curve $y = y(t)$ at $t = t_1$ by the height of the line tangent to $y = y(t)$ at the point (t_0,y_0). To find $y_2 \approx y(t_2)$ replace t_0 by t_1 and y_0 by y_1 in the right member of equation C.6 to obtain :

$$y_2 = y_1 + h.F(t_1,y_1) \qquad (C.6)$$

The process is repeated to find y_i from y_{i-1} for i = 3,4,5,..m by the recursion formula :

$$y_{n+1} = y_n + h.F(t_n,y_n) \qquad (C.7)$$

The error due to the Euler linear approximation at the nth step of the Euler process is given by :

$$e_n = \frac{y_n - y(t_n)}{y(t_n)} . 100\% \qquad (C.8)$$

This error e_n is known as the truncation or discretization error since equation C.7 uses the function F at a discrete number of points, whereas in equation C.2 F varies continuously over the interval $[t_0,t_m]$. The error e_n is said to have order of accuracy h. A further error occurs when using a digital computer to perform these iterations. When calculations are made with real numbers on a digital computer, the results are rounded off to a certain number of decimal points. The smaller h is made, the more calculations have to be made and the calculation error gets larger. This is known as roundoff error. Figure C.3 shows the relationship between the truncation error and roundoff error for different step sizes of h. From this curve it is clear that there is some optimum value of step size h_m that will give the smallest error possible.

APPENDIX D

LISTING OF THE PASCAL MODEL SUBROUTINE

APPENDIX D

THE MODEL SUBROUTINE PASCAL LISTING

```
This routine contains all the model variable declarations and the model
equations. It also contains routines for initialising the model to its
unanaesthetised state and routines for selecting parameters for plotting
and graphing purposes.
}

unit model3;                    {Declared as a TURBO UNIT. Called by the}
                                {main program}
interface

uses math, bartest, printer;    {This routine uses these UNITS as well}

{-------------------------------------------------------------------
*
*                       Model constant declarations.
*
*-------------------------------------------------------------------}

TYPE modelrecord = RECORD

                        BR,         {Breathing rate Breaths/Min}
                        BR0,        {Initial breathing rate Breaths/Min}

                        DiasP,      {Diastolic blood pressure mmHg}

                        fa_fi,      {Alveolar to inspired concentration ratio}

                        Gsum,       {Sum of conductences L/min/mmHg}
                        Gsum0,      {Initial sum of conductances}


                        HR,         {Heart rate B/Min}
                        HR0,        {Initial heart rate}

                        MABP,       {Mean Arterial Blood Pressure mmHg}
                        MABP0,      {Initial Mean Arterial Blood Pressure mmHg}

                        Pamb,       {Atmospheric pressure 760mmHg}
                        Ph20,       {Water vapour pressure 47mmHg}

                        Pins,       {Inspired halothane partial pressure %Atm}
                        Pfr,        {Fresh gas halothane partial pressure %Atm}

                        PulsP,      {Pulse blood pressure mmHg}

                        Q,          {Cardiac output L/Min}
                        Qalv,       {Alveolar minute volume L/Min}
                        Qalv0,      {Initial alveolar minute volume L/min}
                        Qdead,      {Dead space ventilation L/Min}
                        Q0,         {Initial cardiac output L/Min}

                        Qe,         {Ventilation minute volume L/Min}
                        Qfr,        {Fresh gas flow L/Min}

                        Rsum,       {Peripheral resistance mmHg/L/Min}

                        shunt,      {Lung shunt in fraction of cardiac output}
```

```
                    SystP,      {Systolic blood pressure mmHg}

                    Sv,         {Stroke volume L}
                    Sv0,        {Initial stroke volume L}

                    Vbc,        {Breathing circuit volume}

                    Vdead,      {Dead space L}

                    Vfrc,       {Functional residual capacity L}

                    Vtidal,     {Tidal volume L}
                    Vtidal0,    {Initial tidal volume L}

                    Vart,       {Arterial blood volume}
                    Vven,       {Vena Carva blood volume}
                    Vfb,        {Fat blood volume L}
                    Vgb,        {Grey matter blood  volume L}
                    Vhb,        {Heart blood volume L}
                    Vib,        {Viscera blood volume L}
                    Vlb,        {Lung Blood volume L}
                    Vmb,        {Muscle blood volume L}
                    Vpb,        {Poor blood volume L}
                    Vrb,        {Rich blood volume L}
                    Vwb,        {White matter blood volume L}

                    Vft,        {Fat tissue volume L}
                    Vgt,        {Grey matter tissue volume  L}
                    Vht,        {Heart muscle volume L}
                    Vit,        {Viscera tissue volume L}
                    Vlt,        {Lung tissue volume L}
                    Vmt,        {Muscle tissue volume L}
                    Vpt,        {Poorly perfused tissue volume L}
                    Vrt,        {Richly perfused tissue volume L}
                    Vwt,        {White matter tissue volume L}

                    Yb,         {Blood   solubility coefficient}
                    Yf,         {Fat          "            "     }
                    Yg,         {Grey         "            "     }
                    Yh,         {Heart        "            "     }
                    Yi,         {Viscera      "            "     }
                    Yl,         {Lung         "            "     }
                    Ym,         {Muscle       "            "     }
                    Yp,         {Poor         "            "     }
                    Yr,         {Rich         "            "     }
                    Yw,         {White        "            "     }

{-------------------------------------------------------------------
*
*                Model derived variables.
*
*-------------------------------------------------------------------}

                    Af,
                    Abc,
                    Ag,
                    Ah,
                    Ai,
                    Al,     {Amount of anaesthetic in compartment (L)}
                    Am,
                    Ap,
                    Ar,
```

```
Aw,
Aa,
Av,

Capbc,
Capf,
Caph,
Capi,
Capw,
Capg,
Capr,   {Capacity for anaesthetic at 1 atmosphere}
Capp,   {measured in Litres/100                   }
Capl,
Capa,
Capv,
Capm,

dPbcdt,     {1st derivative of compartment anaesthetic}
dPfdt,      {partial pressure}
dPgdt,
dPhdt,
dPidt,
dPldt,
dPmdt,
dPpdt,
dPrdt,
dPwdt,

kbc,        {Breathing circuit constant}
kf,         {Fat constant}
kg,         {Grey matter constant}
kh,         {Heart constant}
ki,         {Viscera constant}
kl,         {Lung constant}
km,         {Muscle constant}
kp,         {Poor constant}
kr,         {Rich constant}
kw,         {White matter constant}

Gf,
Gg,
Gh,
Gi,
Gm,         {Compartment conductences}
Gp,
Gr,
Gsk,
Gw,

Gf0,        {Initial Compartment conductences}
Gg0,        {Res = Pressure/flow = mmHg/L/min}
Gh0,        {Cond = 1/Res = L/min/mmHg       }
Gi0,
Gm0,
Gp0,
Gr0,
Gsk0,
Gw0,

Phfr,       {Fresh halothane conc saturated}

Pa,         {Arterial anaesthetic partial pressure %atm}
```

```
                    Pbc,        {Breathing circuit       "        "       " }
                    Pf,         {Fat             "       "        "       " }
                    Pg,         {Grey matter     "       "        "       " }
                    Ph,         {Heart           "       "        "       " }
                    Pi,         {Viscera         "       "        "       " }
                    Pl,         {Alveolar        "       "        "       " }
                    Pm,         {Muscle          "       "        "       " }
                    Pp,         {Poor            "       "        "       " }
                    Pr,         {Rich            "       "        "       " }
                    Pv,         {Mixed venous    "       "        "       " }
                    Pw,         {White matter    "       "        "       " }

                    Qf,         {Fat             "                    }
                    Qg,         {Grey matter blood flow L/min         }
                    Qh,         {Heart           "                    }
                    Qi,         {Visceral        "                    }
                    Ql,         {Lung            "                    }
                    Qm,         {Muscle          "                    }
                    Qp,         {Poor            "                    }
                    Qr,         {Rich            "                    }
                    Qsk,        {Skin shunt      "                    }
                    Qs,         {Lung shunt      "                    }
                    Qw          {White matter Blood Flow L/min}
                                : REAL;
               END;

{-------------------------------------------------------------
*
*       Declare all plotable parameters as enumerated type.
*
*-------------------------------------------------------------}


TYPE    ptype       =
        (p_strt,
         p_BR,          {Breathing rate Breaths/Min}
         p_DiasP,       {Diastolic blood pressure mmHg}
         p_Gsum,        {Sum of conductences L/min/mmHg}
         p_Rsum,        {Peripheral resistance mmHg/L/Min}
         p_HR,          {Heart rate B/Min}
         p_MABP,        {Mean Arterial Blood Pressure mmHg}
         p_PulsP,       {Pulse blood pressure mmHg}
         p_SystP,       {Systolic blood pressure mmHg}
         p_Sv,          {Stroke volume L}
         p_Vtidal,      {Tidal volume L}
         p_Gf,
         p_Gg,
         p_Gh,
         p_Gi,
         p_Gm,          {Compartment conductences}
         p_Gp,
         p_Gr,
         p_Gsk,
         p_Gw,
         p_Pa,          {Arterial anaesthetic partial pressure %atm}
         p_fa_fi,       {Alveolar to inspired concentration ratio}
         p_Pbc,         {Breathing circuit       "       "       " }
         p_Pf,          {Fat             "       "       "       " }
         p_Pg,          {Grey matter     "       "       "       " }
         p_Ph,          {Heart           "       "       "       " }
         p_Pi,          {Viscera         "       "       "       " }
         p_Pins,        {Inspired concentration halothane Vol%     }
```

```
          p_Pfr,        {Fresh conc                                  }
          p_Pl,         {Alveolar     "          "        "       " }
          p_Pm,         {Muscle       "          "        "       " }
          p_Pp,         {Poor         "          "        "       " }
          p_Pr,         {Rich         "          "        "       " }
          p_Pv,         {Mixed venous "          "        "       " }
          p_Pw,         {White matter "          "        "       " }
          p_Qf,         {Fat          "                 }
          p_Q,          {Cardiac output L/Min}
          p_Qalv,       {Alveolar minute volume L/Min}
          p_Qe,         {Ventilation minute volume L/Min}
          p_Qfr,        {Fresh gas flow rate L/Min}
          p_Qg,         {Grey matter blood flow L/min }
          p_Qh,         {Heart        "                 }
          p_Qi,         {Visceral     "                 }
          p_Ql,         {Lung         "                 }
          p_Qm,         {Muscle       "                 }
          p_Qp,         {Poor         "                 }
          p_Qr,         {Rich         "                 }
          p_Qsk,        {Skin shunt   "                 }
          p_Qw,         {White matter Blood Flow L/min}
          p_Abc,
          p_Af,
          p_Ag,
          p_Ah,
          p_Ai,
          p_Al,         {Amount of gas Litres}
          p_Am,
          p_Ap,
          p_Ar,
          p_Aw,
          p_Aa,
          p_Av,
          p_null);       {Null parameter}

VAR modelparameters : modelrecord;

{-----------------------------------------------------------}

PROCEDURE initialisemodelconstants;
PROCEDURE initialisemodelconstants2;
PROCEDURE initialisemodelstart;
PROCEDURE model(dt : REAL);
PROCEDURE initbarrecord(parm : ptype;
                    VAR b    : barrecord);
FUNCTION getp(parm : ptype) : REAL;
PROCEDURE datacheck;

implementation

{-----------------------------------------------------------}

PROCEDURE initialisemodelconstants;

{
Initialise all model parameters to default unanaesthetised state}
}

BEGIN
WITH modelparameters DO
BEGIN
```

```
    BR0      := 16;
    HR0      :=  70;
    MABP0    := 100;
    Pamb     := 760;
    Ph20     := 47;
    Q0       := 5.8;
    shunt    := 0.03;
    SV0      := Q0/HR0;
    Vdead    := 0.150;
    Vfrc     := 2.5;
    Vtidal0  := 0.375;

    Vbc      := 10;

    Vft      := 12.2;
    Vgt      := 0.75;
    Vht      := 0.3;
    Vit      := 3.9;
    Vlt      := 0.6;
    Vmt      := 33.0;
    Vpt      := 6.2;
    Vrt      := 0.34;
    Vwt      := 0.75;

    Vart     := 0.96;
    Vfb      := 0.160;
    Vgb      := 0.371;
    Vhb      := 0.148;
    Vib      := 0.976;
    Vlb      := 0.372;
    Vmb      := 0.410;
    Vpb      := 0.117;
    Vrb      := 0.876;
    Vwb      := 0.100;
    Vven     := 0.56;

    Yb       := 2.3;
    Yf       := 138;
    Yg       := 5.4;
    Yh       := 8.1;
    Yi       := 6.0;
    Yl       := 5.3;
    Ym       := 8.1;
    Yp       := 5.3;
    Yr       := 3.7;
    Yw       := 8.3;

    Gf0    := 0.002;
    Gg0    := 0.006;
    Gh0    := 0.0025;
    Gi0    := 0.0143;
    Gm0    := 0.012;
    Gp0    := 0.0019;
    Gr0    := 0.0128;
    Gsk0   := 0.005;
    Gw0    := 0.0015;

    Gsum0 := Gg0 + Gw0 + Gr0 + Gm0 + Gsk0 + Gh0 + Gp0 + Gi0 + Gf0;

END;
END;     { initialise constants }
```

```
{-------------------------------------------------------------------}

PROCEDURE initialisemodelconstants2;

{
Calculate the initial values for compartment capacities, conductances
and differential equation constants K
}

{All compartment capacities /100 to speed up calculations in model
 subroutine. .ie Amount anaesthetic in compartment i is

                  Aci := Pci*Capci/100;

The /100 term converts the partial pressure from %Atm to fraction of
atmospheric.                                                        }

BEGIN
WITH modelparameters DO
BEGIN
   kbc := 1/Vbc;                                  { = 0.2500}
   Capbc:= Vbc/100;

   Capf:= (Yb*Vfb + Yf*Vft);
   kf  := Yb/Capf;                                { = 0.0014}
   Capf:= Capf/100;

   Capg:= (Yb*Vgb + Yg*Vgt);
   kg  := Yb/Capg;                                { = 0.4691}
   Capg:= Capg/100;

   Caph:= (Yb*Vhb + Yh*Vht);
   kh  := Yb/Caph;                                { = 0.8302}
   Caph:= Caph/100;

   Capi:= (Yb*Vib + Yi*Vit);
   ki  := Yb/Capi;                                { = 0.0897}
   Capi:= Capi/100;

   Capl:= (Yb*Vlb + Yl*Vlt+Vfrc + Vtidal/2);
   kl  := Yb/Capl;                                { = 0.9119}

   Capm:= (Yb*Vmb + Ym*Vmt);
   km  := Yb/Capm;                                { = 0.0086}
   Capm:= Capm/100;

   Capp:= (Yb*Vpb + Yp*Vpt);
   kp  := Yb/Capp;                                { = 0.5766}
   Capp:= Capp/100;

   Capr:= (Yb*Vrb + Yr*Vrt);
   kr  := Yb/Capr;                                { = 0.7028}
   Capr:= Capr/100;

   Capw:= (Yb*Vwb + Yw*Vwt);
   kw  := Yb/Capw;                                { = 0.3563}
   Capw:= Capw/100;

   Capa:= Yb*Vart/100;

   Capv:= Yb*Vven/100;
```

```
END;
END;     { initialise model }

{-------------------------------------------------------------------------}

PROCEDURE initialisemodelstart;

{
Initialise variable parameters to initial unanaesthetised state
}

BEGIN
WITH modelparameters DO
 BEGIN

   Pfr    := 0;
   Pins   := 0;

   Pa     := 0;
   Pbc    := 0;
   Pf     := 0;
   Pg     := 0;
   Ph     := 0;
   Pi     := 0;
   Pl     := 0;
   Pm     := 0;
   Pp     := 0;
   Pr     := 0;
   Pv     := 0;
   Pw     := 0;

   Abc    := 0;
   Al     := 0;
   Ag     := 0;
   Aw     := 0;
   Ah     := 0;
   Ai     := 0;
   Ap     := 0;
   Ar     := 0;
   Af     := 0;
   Am     := 0;
   Aa     := 0;
   Av     := 0;

   Q      := Q0;
   Qfr    := 8;

   Ql     := (1-shunt)*Q0;
   Qs     := shunt*Q0;
   Qg     := MABP0 * Gg0;
   Qw     := MABP0 * Gw0;
   Qi     := MABP0 * Gi0;
   Qr     := MABP0 * Gr0;
   Qm     := MABP0 * Gm0;
   Qh     := MABP0 * Gh0;
   Qp     := MABP0 * Gp0;
   Qf     := MABP0 * Gf0;
   Qsk    := MABP0 * Gsk0;

   Gg     := Gg0;
   Gw     := Gw0;
```

```
   Gi     := Gi0;
   Gr     := Gr0;
   Gp     := Gp0;
   Gm     := Gm0;
   Gh     := Gh0;
   Gf     := Gf0;
   Gsk    := Gsk0;

   Gsum   := Gsum0;
   Rsum   := 1/Gsum0;

   MABP   := MABP0;
   HR     := HR0;;

   Sv     := Sv0;

   PulsP  := 40 * Sv0/Sv;
   SystP  := MABP0 + 2*PulsP/3;
   DiasP  := MABP0 - PulsP/3;

   BR     := BR0;
   Vtidal:= Vtidal0;

   Qe     := BR*Vtidal;
   Qalv   := BR*(Vtidal - Vdead);
   Qdead  := Qe - Qalv;

   fa_fi  := 0;

 END;
END;     { initialise start }

{-------------------------------------------------------------------------}

PROCEDURE model(dt : REAL);

{
 This is a series of differential and linear equations that describe the
 way in which various model parameters change in relation to one another.
 This subroutine is iterated repeatedly over the time interval dt
}

BEGIN  { model }

WITH modelparameters DO
BEGIN

{Breathing circuit equations}

   Qe    := BR * Vtidal;
   Qalv  := BR*(Vtidal - Vdead);
   Qdead:= Qe-Qalv;

{Breathing circle based on Egers rebreathing circuit model H}

   Phfr  := Pfr*(Pamb/(Pamb+Ph20));    {Adjust Pfr to BTPS FROM ATPD}

   IF Qfr > Qe THEN
   BEGIN
      Pins  := Phfr;
      Pbc   := Phfr;
   END
```

```
   ELSE
    BEGIN
       IF Qalv < Qfr THEN
          dPbcdt := (Qe-Qfr)*(Pins-Pbc)/Vbc
      ELSE
          dpbcdt := (Qdead*(Pins-Pbc) + Qalv*(Pl-Pbc))/Vbc;
       Pbc := Pbc + dPbcdt*dt;
       Abc := (Pbc/100)*Vbc;
       Pins := Phfr*(Qfr/Qe) + Pbc*(Qe-Qfr)/Qe;
    END;

{Lung compartment equations}

   Ql := (1-shunt)*Q;
   Capl := (Yb*Vlb + Yl*Vlt+Vfrc + Vtidal/2);
   kl := Yb/Capl;
   dPldt := kl*(Ql*(Pv - Pl) + Qalv*(Pbc - Pl)/Yb);
   Pl := Pl + dPldt*dt;
   Al := (Pl/100)*Capl;

{Lung shunt equations}

   Qs := shunt*Q;
   Pa := (Qs*Pv + Ql*Pl)/Q;
   Aa := Pa*Capa;

{Cerebral grey mater equations}

   Gg := Gg0*(1 + 0.75*Pa);
   IF Gg > 0.015 THEN Gg := 0.015;
   Qg := MABP*Gg;
   dPgdt := Qg*kg*(Pa - Pg);
   Pg := Pg + dPgdt*dt;
   Ag := Pg*Capg;

{Cerebral white mater equations}

   Gw := Gw0*(1 + 0.7533*Pa);
   IF Gw > 0.0038 THEN Gw := 0.0038;
   Qw := MABP*Gw;
   dPwdt := Qw*kw*(Pa - Pw);
   Pw := Pw + dPwdt*dt;
   Aw := Pw*Capw;

{Richly perfused tissue equations}

   Gr := Gr0*(1 - 0.2836*Pa);
   IF Gr < 0.00554 THEN Gr := 0.00554;
   Qr := MABP*Gr;
   dPrdt := Qr*kr*(Pa - Pr);
   Pr := Pr + dPrdt*dt;
   Ar := Pr*Capr;

{Muscle tissue equations}

   Gm := Gm0*(1 - 0.1*Pa);
   IF Gm < 0.0096 THEN Gm := 0.0096;
   Qm := MABP*Gm;
   dPmdt := Qm*km*(Pa - Pm);
   Pm := Pm + dPmdt*dt;
   Am := Pm*Capm;
```

```
{Heart muscle and myocardium equations}

   Qh := MABP*Gh0;
   dPhdt := Qh*kh*(Pa - Ph);
   Ph := Ph + dPhdt*dt;
   Ah := Ph*Caph;

{Poorly perfused tissue equations}

   Qp := MABP*Gp0;
   dPpdt := Qp*kp*(Pa - Pp);
   Pp := Pp + dPpdt*dt;
   Ap := Pp*Capp;

{Splanchnic bed equations}

   Qi := MABP*Gi0;
   dPidt := Qi*ki*(Pa - Pi);
   Pi := Pi + dPidt*dt;
   Ai := Pi*Capi;

{Fatty tissue equations}

   Qf := MABP*Gf0;
   dPfdt := Qf*kf*(Pa - Pf);
   Pf := Pf + dPfdt*dt;
   Af := Pf*Capi;

{Skin shunt equations}

   Gsk := Gsk0*(1 + Pa);
   IF Gsk > 0.015 THEN Gsk := 0.015;
   Qsk := MABP*Gsk;

{Venous blood equation}

   Pv := (Qg*Pg+Qw*Pw+Qr*Pr+Qm*Pm+Qh*Ph+Qp*Pp+Qi*Pi+Qf*Pf+Qsk*Pa)/Q;
   Av := Pv*Capv;

{Total conductance}

   Gsum := Gg + Gw + Gr + Gm + Gsk + Gh0 + Gp0 + Gi0 + Gf0;
   Rsum := 1/Gsum;

{Cardiac output as a function of intestinal halothane concentration}

   Q := Q0*(1-0.2*Ph);
   Sv:= Sv0*(1-0.25*Ph);

{Mean arterial blood pressure}

   MABP := Q/Gsum;

{Heart rate baro-receptor relationship}

   hr := Q/SV;

{Pulsatile blood pressure relationships}

   PulsP := 40 * Sv/Sv0;
   SystP := MABP + 2*PulsP/3;
   DiasP := MABP - PulsP/3;
```

```
{Alveolar to inspired concentration ratio}

   IF Pins > 0 THEN fa_fi := (Pl/Pins)*100 ELSE fa_fi := 0;

END;
END;     { model }

{-------------------------------------------------------------------------}

PROCEDURE initbarrecord(parm : ptype;
                        VAR b    : barrecord);

{
Get details about a plotable parameter such as its title, maximum value
for a bargraph, and the units its measured in
}

BEGIN
   WITH b DO
   BEGIN
      min := 0;
   CASE parm OF
      p_BR      : BEGIN max:=20;units:='L/M';title:='BREATHING RATE';END;
      p_DiasP   : BEGIN max:=150;units:='mmHg';title:='DIASTOLIC BP';END;
      p_fa_fi   : BEGIN max:=100;units:='';title:='ALV/INS RATIO';END;
      p_Gsum    : BEGIN max:=0.1;units:='';title:='CONDUCTANCE';END;
      p_Rsum    : BEGIN max:=10;units:='';title:='RESISTANCE';END;
      p_HR      : BEGIN max:=100;units:='B/MIN';title:='HEART RATE';END;
      p_MABP    : BEGIN max:=150;units:='mmHg';title:='BLOOD PRESSURE';END;
      p_Pins    : BEGIN max:=4;units:='%ATM';title:='INS HALOTHANE';END;
      p_Pfr     : BEGIN max:=4;units:='%ATM';title:='FRESH HALOTHANE';END;
      p_PulsP   : BEGIN max:=150;units:='mmHg';title:='PULSE PRESSURE';END;
      p_Q       : BEGIN max:=6;units:='L/MIN';title:='CARDIAC OUTPUT';END;
      p_Qe      : BEGIN max:=10;units:='L/MIN';title:='VENT MINUTE VOL';END;
      p_Qalv    : BEGIN max:=10;units:='L/MIN';title:='ALVEOLAR VENT';END;
      p_Qfr     : BEGIN max:=10;units:='L/MIN';title:='FRESH GAS FLOW';END;
      p_SystP   : BEGIN max:=150;units:='mmHg';title:='SYSTOLIC BP';END;
      p_Sv      : BEGIN max:=0.25;units:='L';title:='STROKE VOLUME';END;
      p_Vtidal  : BEGIN max:=1;units:='L';title:='TIDAL VOLUME';END;
      p_Gf      : BEGIN title:='FAT   COND';END;
      p_Gg      : BEGIN title:='GRAY  COND';END;
      p_Gh      : BEGIN title:='HEART COND';END;
      p_Gi      : BEGIN title:='VISCERA COND';END;
      p_Gm      : BEGIN title:='MUSCLE COND';END;
      p_Gp      : BEGIN title:='POOR COND';END;
      p_Gr      : BEGIN title:='RICH COND';END;
      p_Gsk     : BEGIN title:='SKIN COND';END;
      p_Gw      : BEGIN title:='WHITE COND';END;
      p_Pa      : BEGIN title:='ARTERIAL PP';END;
      p_Pbc     : BEGIN title:='BREATH CRCT PP';END;
      p_Pf      : BEGIN title:='FAT PP';END;
      p_Pg      : BEGIN title:='GRAY PP';END;
      p_Ph      : BEGIN title:='HEART PP';END;
      p_Pi      : BEGIN title:='VISCERA PP';END;
      p_Pl      : BEGIN title:='LUNG PP';END;
      p_Pm      : BEGIN title:='MUSCLE PP';END;
      p_Pp      : BEGIN title:='POOR PP';END;
      p_Pr      : BEGIN title:='RICH PP';END;
      p_Pv      : BEGIN title:='VENOUS PP';END;
      p_Pw      : BEGIN title:='WHITE PP';END;
      p_Qf      : BEGIN title:='FAT BF';END;
```

```
      p_Qg     : BEGIN title:='GRAY BF';END;
      p_Qh     : BEGIN title:='HEART BF';END;
      p_Qi     : BEGIN title:='VISCERAL BF';END;
      p_Ql     : BEGIN title:='LUNG BF';END;
      p_Qm     : BEGIN title:='MUSCLE BF';END;
      p_Qp     : BEGIN title:='POOR BF';END;
      p_Qr     : BEGIN title:='RICH BF';END;
      p_Qsk    : BEGIN title:='SKIN BF';END;
      p_Qw     : BEGIN title:='WHITE BF';END;
      p_Af     : BEGIN title:='FAT AMOUNT';END;
      p_Abc    : BEGIN title := 'B CRCT AMOUNT'; END;
      p_Ag     : BEGIN title := 'GREY AMOUNT';END;
      p_Ah     : BEGIN title := 'HEART AMOUNT';END;
      p_Ai     : BEGIN title := 'VISCERAL AMNT';END;
      p_Al     : BEGIN title := 'LUNG AMOUNT';END;
      p_Am     : BEGIN title := 'MUSCLE AMOUNT';END;
      p_Ap     : BEGIN title := 'POOR AMOUNT';END;
      p_Ar     : BEGIN title := 'RICH AMOUNT';END;
      p_Aw     : BEGIN title := 'WHITE AMOUNT';END;
      p_Aa     : BEGIN title := 'ARTERIAL AMOUNT';END;
      p_Av     : BEGIN title := 'VENOUS AMOUNT';END;
      p_null   : BEGIN title:='NOTHING'; END;
   END; {Case}
   END;
END;

{-------------------------------------------------------}

FUNCTION getp(parm : ptype) : REAL;

{
Get the actual value of a plottable parameter for graphing purposes
}

BEGIN
   WITH modelparameters DO
   CASE parm OF
         p_BR    : getp := BR;
         p_DiasP : getp := diasp;
         p_fa_fi : getp := fa_fi;
         p_Gsum  : getp := Gsum;
         p_Rsum  : getp := Rsum;
         p_HR    : getp := HR;
         p_MABP  : getp := MABP;
         p_Pins  : getp := Pins;
         p_Pfr   : getp := Pfr;
         p_PulsP : getp := PulsP;
         p_Q     : getp := Q;
         p_Qalv  : getp := Qalv;
         p_Qfr   : getp := Qfr;
         p_Qe    : getp := Qe;
         p_SystP : getp := SystP;
         p_Sv    : getp := Sv;
         p_Vtidal: getp := Vtidal;
         p_Gf    : getp := Gf;
         p_Gg    : getp := Gg;
         p_Gh    : getp := Gh;
         p_Gi    : getp := Gi;
         p_Gm    : getp := Gm;
         p_Gp    : getp := Gp;
         p_Gr    : getp := Gr;
         p_Gsk   : getp := Gsk;
```

```
            p_Gw     : getp := Gw;
            p_Pa     : getp := Pa;
            p_Pbc    : getp := Pbc;
            p_Pf     : getp := Pf;
            p_Pg     : getp := Pg;
            p_Ph     : getp := Ph;
            p_Pi     : getp := Pi;
            p_Pl     : getp := Pl;
            p_Pm     : getp := Pm;
            p_Pp     : getp := Pp;
            p_Pr     : getp := Pr;
            p_Pv     : getp := Pv;
            p_Pw     : getp := Pw;
            p_Qf     : getp := Qf;
            p_Qg     : getp := Qg;
            p_Qh     : getp := Qh;
            p_Qi     : getp := Qi;
            p_Ql     : getp := Ql;
            p_Qm     : getp := Qm;
            p_Qp     : getp := Qp;
            p_Qr     : getp := Qr;
            p_Qsk    : getp := Qsk;
            p_Qw     : getp := Qw;
            p_Af     : getp := Af;
            p_Abc    : getp := Abc;
            p_Ag     : getp := Ag;
            p_Ah     : getp := Ah;
            p_Ai     : getp := Ai;
            p_Al     : getp := Al;
            p_Am     : getp := Am;
            p_Ap     : getp := Ap;
            p_Ar     : getp := Ar;
            p_Aw     : getp := Aw;
            p_Aa     : getp := Aa;
            p_Av     : getp := Av;
            p_null   : getp := -99;
        END; {Case}
END;

{-------------------------------------------------------}

PROCEDURE datacheck;

{
Test procedure used during model development. Not used by main
program.
}

VAR lst : TEXT;

BEGIN
   ASSIGN(lst,'');
   REWRITE(lst);
   WITH modelparameters DO
   BEGIN
      WRITELN(lst,'Compart':9,'Const':8,'Cap/100':8,'Flow0':8);
      WRITELN(lst,'Brcrct   ',kbc:8:4,Vbc/100:8:4,Qalv:8:4);
      WRITELN(lst,'Fat      ',Kf:8:4,Capf:8:4,Qf:8:4);
      WRITELN(lst,'Gray     ',Kg:8:4,Capg:8:4,Qg:8:4);
      WRITELN(lst,'Heart    ',Kh:8:4,Caph:8:4,Qh:8:4);
      WRITELN(lst,'Viscera  ',Ki:8:4,Capi:8:4,Qi:8:4);
      WRITELN(lst,'Lung     ',Kl:8:4,Capl:8:4,Ql:8:4);
```

```
      WRITELN(lst,'Muscle    ',Km:8:4,Capm:8:4,Qm:8:4);
      WRITELN(lst,'Poor      ',Kp:8:4,Capp:8:4,Qp:8:4);
      WRITELN(lst,'Rich      ',Kr:8:4,Capr:8:4,Qr:8:4);
      WRITELN(lst,'White     ',Kw:8:4,Capw:8:4,Qw:8:4);
      WRITELN(lst,'Arterial  ','':8,Capa:8:4,Q:8:4);
      WRITELN(lst,'Venous    ','':8,Capv:8:4,Q:8:4);
      WRITELN(lst);
   END;
END;

{-------------------------------------------------------}

BEGIN
END.
```

APPENDIX E

A DESCRIPTION OF TURBO PASCAL V4.0 UNITS CALLED BY THE SIMULATOR PROGRAM

APPENDIX E

A DESCRIPTION OF THE ROUTINES RESIDING IN PRE-COMPILED TURBO V4.0 UNITS THAT ARE CALLED BY THE SIMULATOR PROGRAM.

E.1 THE TURBO GRAPHIX TOOLBOX UNITS.

This graphics package consists of three units that contain all the basic graphics routines :

Gdriver - Contains graphics card information. In the case of the simulator it has been set up to talk to a hercules or monochrome graphics card.
modkern - Contains the basic graphics routines.
Gwindow - Contains routines for manipulating windows.

The hercules card has the following characteristics :
Screen size in graphics mode = x - 0 to 719
y - 0 to 299
0,0, is the top left corner in screen coordinates

INITIALISING GRAPHICS.

To initialise and leave graphics mode the following routines are called :

initgraphic; - initialise the graphics system.
leavegraphic; Enter textmode. Defined windows and worlds are preserved.
entergraphic; Reenter graphics mode after a leavegraphic has been called. Old windows and worlds can still be used.

WINDOWS AND WORLDS.

Windows are defined in screen coordinates as follows :
definewindow(win,x1,y1,x2,y2); where win is the window number and x1,y1 = the top left coord and x2,y2 = the lower right.
NB!!! The x screen coordinates can only be declared at positions that are a multiple of 8. ie Divide actual screen position by 8 and round of to nearest integer for x values. eg screen x position 121 is declared as 15 in definewindow procedure.

Windows can have a world coordinate system assigned to them as follows :
First define a world system as :
defineworld(world,x1,y1,x2,y2); where world is the world number, x1,y1 refer to the lower left world coordinate and x2,y2 the top right.
eg. If one wants to plot data that ranges from -5 to 6 over the x axis and from 0 to 0.5 over the y axis in a window which covers the whole graphics screen then proceed as follows :

```
definewindow(win,0,0,90,299);  window over whole screen
                               NB! x2 = 90 = position 720/8
defineworld(world,-5,0,6,0.5); define world coordinates
```

```
selectworld(world);            select the world
selectwindow(win);             select the window
```
Data is now plotted in this window according to the world coordinate system world.

A LIST OF ROUTINES USED BY THE SIMULATOR.

Routines used by the simulator are as follows :

```
initgraphic;
leavegraphic;
entergraphic;
definewindow(win,x1,y1,x2,y2);
defineworld(world,x1,y1,x2,y2);
selectwindow(win);
selectworld(world);

drawborder;                  - draw a border around the
                               current window
setbackground(0);            - clear the current window
setlinestyle(style);         - Select a linestyle 0-255.
                               0 = solid line - the rest
                               are bit mapped
drawtextw(x,y,size,string); - draw text string at world
                               coordinate x,y of of size
                               size. The 4X6 size font is
                               used.
drawline(x1,y1,x2,y2);       - draw a line from world
                               coordinate x1,y1 to x2,y2.
                               Uses the current linestyle.
storewindow(win);            - save a window on the window
                               stack for later retrieval.
restorewindow(win,x,y);      - restore a saved window from
                               the window stack. x,y is an
                               ofset in that is ignored if
                               they are negative.
copyscreen;                  - Copy the inactive screen to
                               the active one.
swapscreen;                  - Swap the inactive screen with
                               the active one
selectscreen(screen);        - select the visual or the ram
                               screen as the active screen.
                               screen = 0 is the visual
                               screen, 1 = the ram screen.
                               All plotting and drawing takes
                               place in the active screen.
gotoxy(x,y);                 - Send the cursor to text
                               coordinate x,y. x = 1 to 80
                                               y = 1 to 25
                               0,0 = top left.
WRITE OR WRITELN(data);      - Write text at current text
                               coordinate. Uses 12X9 size
                               font. Data can be number or
                               string. If the global variable
                               invglb is TRUE then the text
                               is written in inverse color.
```

E.2 THE BARTEST UNIT.

Two routines are called from this unit by the main program :

```
Procedure Normbar(VAR bargraph : barrecord; Value : REAL);
and
Procedure Sidebar(VAR bargraph : barrecord; Value : REAL);
```

where normbar draws a vertical bargraph and sidebar draws a horizontal bargraph with the maximum value on the right. Both of these routines require a variable of type barrecord to be passed to them. The record has the following format :

```
TYPE barrecord = RECORD
                    x1,y1,x2,y2 : REAL;
                    max,min     : REAL;
                    len,plc,td  : INTEGER;
                    units,title : STRING;
                    side        : CHAR;
                    frame       : BOOLEAN;
                 END;
```

x1,y1 = Top left graph frame coordinates
x2,y2 = bottom right frame coordinates
Coordinates are in world coordinates as defined with the defineworld procedure from TURBO GRAPHIX TOOLBOX.

max,min
= Maximum and minimum values to be plotted in the bargraph frame

Len,plc,td
= Field length, decimal places of numbers labeling the bargraph axes and the tick density of markers.

Units,title
= Strings to identify the bargraph.

side = Side of graph to plot the title. (L)eft, (R)ight, (T)op, or (B)ottom for normbar and sidebar respectively.

Frame = If FALSE then the bargraph frame is drawn before the data is plotted and then set TRUE. If TRUE then the bargraph is updated with the variable value.

E.3 THE AXES UNIT.

The axes unit is used inconjunction with the TURBO GRAPHIX TOOLBOX. Available routines are :

```
PROCEDURE drawaxes(win,wrld             : INTEGER;
                   xtick,ytick          : REAL;
                   xaxis,yaxis,
                   title                : string;
                   xlen,xpl,ylen,ypl    : INTEGER;
                   id                   : BOOLEAN);
```

The parameters passed to drawaxes are defined as :

win - Predeclared window to draw fraph in.
wrld - Predeclared world (ie 0,0 to 10,100).
xtick,
ytick - Tick density along x and y axis.
xaxis,
yaxis,
title - Axis titles and main title of graph.
xlen,

```
        xpl,
        ylen,
        ypl    - Field length and dec places of axis numbers.
```
*Note drawaxes will only work in graphics mode setup by turbographics toolbox.

```
PROCEDURE idgraph(win,number,size : INTEGER);
```
*If id was set TRUE during drawaxes call then idgraph will identify up to 6 variables whose names must be in the array lineid. The names are drawn in the space left under the x axes.
Each line number is identified with a unique character every maxcount points that are plotted. Idgraph draws these characters next to the name as well.

```
PROCEDURE plotxy(x,y          : REAL;
                 line,
                 linestyle    : INTEGER);
```
*Plots line of linetype from ox,oy to x,y. If oxyinit[line] is false then saves x,y in ox,oy for that line and sets oxyinit[line] true.

```
PROCEDURE plotpoints(x,y : REAL; line,linetype : INTEGER);
```
*Plots line of linetype from ox,oy to x,y. If oxyinit[line] is false then saves x,y in ox,oy for that line and sets oxyinit[line] true.
If linecount[line] is > than maxcount it is reset to zero and a character that identifies that line is drawn at x,y as well.

E.4 THE BSELECT UNIT.

This unit has one routine that is called by the simulator main program. It is called as follows :

```
PROCEDURE selectwithbar(x0,y0,xnum,entrylen,barlen,totentries : INTEGER;
                    VAR  choice : INTEGER;
                    VAR  esc    : BOOLEAN);
```

The parameters are defined as :

```
x0,y0   - The top left text coordinate position of the array
          of strings.
xnum    - The number of entries across top of the string
          array.
entrylen
        - The number of characters in each entry.
barlen  - the length of the movable bar.
totentries
        - The total number of entries in the string array.
choice  - The position in the array that was selected.
esc     - True if escape was pressed.
```

The routine is used to move a bar around on the screen over a defined area of the screen. The programmer must ensure that the correct data strings are written into this area before the routine is called. The bar is moved about with the cursor keys and the return key pressed when a selection is reached. The position of the choice is returned in the variable choice else esc is returned true if escape was pushed.

E.5 VARIOUS OTHER ROUTINES.

The following routines are also called by the main simulator program. The following list indicates what the routines do and which UNIT they reside in :

UNIT GENERAL.

```
Beep(frequency)     - Sounds a tone of frequency hertz for 250
                      ms.
```

UNIT IO.

```
Message(y,string); - Center the message string on line y but
                     clear the line first.
Center(y,string);  - Same as above bu does not clear the
                     line.

kbdstatus    -  The function returns the keyboard status
                as follows :

                ---------------------------------
                | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
                ---------------------------------
                  |   |   |   |   |   |   |   |_ Right shift
                  |   |   |   |   |   |   |_Left shift
                  |   |   |   |   |   |_Cntrl shift
                  |   |   |   |   |_Alt shift
                  |   |   |   |_Scroll lock on
                  |   |   |_Num lock on
                  |   |_Caps lock on
                  |_Insert mode on
```

UNIT CONVERT.

```
hms(time,h,m,s) - Converts a real number time into hour,
                  minute, second format as h,m,s.
```

E.6 THE HLP UNIT.

The help UNIT contains the following three routines :

```
inithelp(filename);
help(filename);
closehelp
```

A call to inithelp will load the help file specified by filename. If the file is not found the program is terminated with an appropriate error message. If there is not enough memory to load the help file into the program is halted with an error message.

If help is called before inithelp has been called, the subroutine calls inithelp itself and then continues. Help saves the contents of the existing text screen, and displays a menu of help topics on the top 10 lines of the screen. The operator moves from topic to topic with the cursor keys. The text describing the currently selected topic is displayed in the lower 15 lines of the screen. The operator can return to the calling routine by pressing escape whereapon the old screen will be restored.

The format of the help file is as follows :

line 1 = Number of help screens to be used.
line 2 = First line of top screen. Help line ie esc, cursor directions
line 3 = Blank
line 4 = First line with 4 topics of 16 characters each.
to
line 10 = Last line of 4 topics.
line 11 = Blank

The rest of the file consists of the help screens each consisting of exactly 14 lines of text . An editor such as PC WRITE can be used to edit the file.

APPENDIX F

LISTING OF THE ERROR CALCULATION PROGRAM

APPENDIX F

LISTING OF THE ERROR CALCULATION PROGRAM

```
{-------------------------------------------------------

        EULER INTEGERATION ERROR CALCULATION PROGRAM
                        RA COOPER
                    CAPE TOWN  1988

-------------------------------------------------------}

uses crt, axes, convert, bartest,          {Declare units}
     GDriver, modkern;

TYPE timearray = ARRAY[1..5] OF barrecord;

{Define actual time bargraph positions and frames}

CONST timebar : timearray = (
      (x1:15; y1:20; x2:21; y2:90;
       max:20; min:0; len:1; plc:0; td:10;
       units:'1X'; title:'SECONDS';
       side:'r'; frame:false),

       (x1:36; y1:20; x2:42; y2:90;
       max:20; min:0; len:1; plc:0; td:10;
       units:'2X'; title:'SECONDS';
       side:'r'; frame:false),

       (x1:57; y1:20; x2:63; y2:90;
       max:20; min:0; len:1; plc:0; td:10;
       units:'4X'; title:'SECONDS';
       side:'r'; frame:false),

       (x1:78; y1:20; x2:84; y2:90;
       max:20; min:0; len:1; plc:0; td:10;
       units:'10X'; title:'SECONDS';
       side:'r'; frame:false),

       (x1:99; y1:20; x2:105; y2:90;
       max:20; min:0; len:1; plc:0; td:10;
       units:'20X'; title:'SECONDS';
       side:'r'; frame:false)
                            );

VAR Pins,           {Inspired halothane %Atm}
    Pn0,            {Initial compartment halothane %Atm}
    a,              {Temp variable for actual value}
    e,              {Temp variable for estimated value}
    dpndt,          {Derivative of compartment concentration}
    emax,           {Maximum error over interval}
    dt,             {Time interval in minutes}
    K,              {Compartment capacity constant}
    Q,              {Compartment blood flow rate - L/Min}
    Z
         : REAL;

    starttime,
    endtime         : LONGINT;

    time,
```

```
    speed,
    i,pnt,
    itterate,
    maxseconds,
    maxitterate     : INTEGER;
    data            : ARRAY[1..8,1..31] OF real;
    filename        : string;
    fil             : text;

{-------------------------------------------------------}

FUNCTION est(e : REAL) : REAL;

{
Estimation of function
}

BEGIN
   est := K*Q*(Pins - e);
END;

{-------------------------------------------------------}

FUNCTION Pact(time : INTEGER) : REAL;

{
Actual value of function
}

VAR Ptemp : REAL;

BEGIN
   Ptemp := Pins + (Pn0 - Pins)*exp(-k*q*time/60);
   Pact := Ptemp;
END;

{-------------------------------------------------------}

FUNCTION error(act,est : REAL) : REAL;

{
Error due to difference between act and est
}

VAR terr : REAL;

BEGIN
   IF act > 0 THEN terr := (ABS(est - act)/act)*100
  ELSE terr := 0;
  error := terr;
END;

{-------------------------------------------------------}

{
Main program
}

VAR count,
    line,
    sm,
    stepsize    : INTEGER;
```

```
    maxerror     : REAL;
    st           : string;
    ch           : CHAR;

BEGIN
   clrscr;
   WRITE(' Enter filename...>');
   READLN(filename);
   filename := 'B:'+filename+'.prn';
   ASSIGN(fil,filename);
   REWRITE(fil);
   WRITE(' Enter speed......>');
   READLN(speed);
   WRITE(' Enter maxerror...>');
   READLN(maxerror);
   WRITE(' Enter max minutes.>');
   READLN(maxseconds);
   maxseconds := maxseconds * 60;
   WRITE(' Elapsed time max >');
   READLN(emax);
   WRITE(' Enter string  ...>');
   READLN(st);
   WRITE(' Enter Pins    ...>');
   READLN(Pins);
   WRITE(' Enter P0      ...>');
   READLN(Pn0);
   WRITE(' Enter (G)rey or (M)uscle ...>');
   ch := READKEY;
   CASE ch OF
      'g','G' : BEGIN
                   {Gray matter}
                   K := 0.4691;
                   Q := 0.6;
                   st := st + ' (Gray matter)';
                END;
      'm','M' : BEGIN
                   {Muscle}
                   K := 0.0086;
                   Q := 1.2;
                   st := st + ' (Muscle)';
                END;
     'r','R'  : BEGIN
                   {Richly perfused tissue}
                   K := 0.7028;
                   Q := 1.28;
                   st := st + ' (Richly perfused)';
                END;
   END;
   st := st + ' '+ist(speed)+'X';

   initgraphic;
   definewindow(15,0,0,xmaxglb,ymaxglb-25);

   definewindow(5,3*xmaxglb DIV 5 ,20,xmaxglb-5,ymaxglb DIV 2);
   definewindow(3,xmaxglb DIV 4,(5*ymaxglb DIV 8) + 10,3*xmaxglb DIV 4,ymaxglb - 10

   defineworld(1,0,0,maxseconds DIV 60,2);
   defineworld(2,0,0,maxseconds DIV 60,maxerror);
   defineworld(5,0,0,120,100);

   selectworld(1);
   selectwindow(15);
```

```
drawaxes(15,1,10,0.5,'MINUTES','CONC HALOTHANE','1 Itt/step Curve',1,0,1,2,true)

lineid[0] := 'ACTUAL';
lineid[1] := 'ESTIMATED';
idgraph(15,2,1);

itterate := 1;
maxcount := round(maxseconds/(10*speed));

drawtextw(0,-0.17,1,'INSPIRED = '+rst(Pins,1,2)+'%; INITIAL = '+rst(Pn0,1,2)+'%'

BEGIN
   oxyinit[0] := FALSE;
   oxyinit[1] := FALSE;
   linecount[0] := 0;
   linecount[1] := maxcount DIV 2;
   dt := speed/(itterate*60);
   e := Pn0;
   plotpoints(0,Pn0,0,0);
   plotpoints(0,Pn0,1,1);
   time := 1*speed;
   WHILE time <= maxseconds DO
   BEGIN
      IF keypressed THEN halt;
      a := pact(time);
      FOR i := 1 TO itterate DO
      BEGIN
         e := e + est(e)*dt;
      END;
      plotpoints(time/60,a,0,0);
      plotpoints(time/60,e,1,1);
      time := time + speed;
   END;
END;
gotoxy(1,25);
WRITELN('P-Print  Space for other screen   Q-Quit');
copyscreen;
clearscreen;

selectworld(2);
selectwindow(15);
drawaxes(15,2,10,maxerror/10,'MINUTES','% ERROR',st,1,0,1,2,true);
selectworld(5);
selectwindow(5);
setbackground(0);
drawborder;
FOR i := 1 TO 5 DO
BEGIN
   timebar[i].max := emax;
   normbar(timebar[i],0);
END;
drawtextw(15,12,1,'ACTUAL TIME TO SIMULATE 30 MINUTES');
drawtextw(15,5,1,'OF DATA AT VARIOUS ITTERATIONS/STEP');

selectworld(2);
selectwindow(15);

lineid[0] := '1 Itt/step';
lineid[1] := '2 Itt/step';
lineid[2] := '4 Itt/step';
lineid[3] := '10 Itt/step';
lineid[4] := '20 Itt/step';
```

```
idgraph(15,5,1);

itterate := 1;
line := 0;
WHILE itterate <= 20 DO
BEGIN
   case itterate OF
      1  : pnt := 2;
      2  : pnt := 3;
      4  : pnt := 4;
      10 : pnt := 5;
      20 : pnt := 6;
   END;
   oxyinit[line] := FALSE;
   dt := speed/(itterate*60);
   e := Pn0;
   plotpoints(0,0,line,0);
   time := 1*speed;
   starttime := tickcount;
   WHILE time <= maxseconds DO
   BEGIN
      IF keypressed THEN halt;
      a := pact(time);
      FOR i := 1 TO itterate DO
      BEGIN
         e := e + est(e)*dt;
      END;
      plotpoints(time/60,error(a,e),line,0);
      IF time mod 60 = 0 THEN
      BEGIN
         data[1,time DIV 60] := time DIV 60;
         data[pnt,time DIV 60] := error(a,e);
         IF pnt = 2 THEN
         BEGIN
            data[7,time DIV 60] := a;
            data[8,time DIV 60] := e;
         END;
      END;
      time := time + speed;
   END;
   endtime := tickcount;
   selectworld(5);
   selectwindow(5);
   normbar(timebar[pnt-1],(endtime - starttime)/tickspersec);
   selectworld(2);
   selectwindow(15);
   data[pnt,31] := (endtime-starttime)/tickspersec;
   IF itterate = 1 THEN itterate := 2
   ELSE
   IF itterate = 2 THEN itterate := 4
   ELSE
   IF itterate = 4 THEN itterate := 10
   else
   IF itterate = 10 THEN itterate := 20
   ELSE itterate := 50;
   line := line + 1;
   IF odd(line) THEN linecount[line] := 1 ELSE
   linecount[line] := 0;
END;
gotoxy(1,25);
WRITELN('P-Print  Space for other screen   Q-Quit  File = ',filename);
REPEAT
```

```
      ch := readkey;
      IF upcase(ch) IN ['p','P'] THEN hardcopy(false,1);
      IF ch = ' ' THEN swapscreen;
   UNTIL ch IN ['q','Q'];
   leavegraphic;
   WRITELN(fil,'"Time  " " 1X      " " 2X       " "4X       " "10X     " " 20X   "
   FOR time := 1 TO 30 DO
   BEGIN
      WRITE(fil,time:6,' ');
      FOR pnt := 2 TO 8 DO
         WRITE(fil,data[pnt,time]:10:7,' ');
      WRITELN(fil);
   END;
   WRITE(fil,'"Elapsed time"');
   FOR i := 2 TO 6 DO WRITE(fil,data[i,31]:10:7,' ');
   WRITELN(fil);
   close(fil);
END.
```

APPENDIX G

LISTING OF THE SIMULATOR PASCAL PROGRAM

APPENDIX G

LISTING OF THE SIMULATOR PASCAL PROGRAM

```
{-------------------------------------------------------
*          UNIVERSITY OF CAPE TOWN ANAESTHETIC TRAINING SIMULATOR.
*                              (UCTAS V2.0)
*          Devevloped at the department of Biomedical Engineering.
*                            University of Cape Town
*                                 By RA Cooper
*                                     1988
*
*-------------------------------------------------------
*
*   Declare all external units used by the program
*
*-------------------------------------------------------}

uses crt, general, io, hlp, model3, convert, axes,printer,
     GDriver, modkern, GWindow, bartest, bselect, gameport;

{-------------------------------------------------------
*
*     UCTAS Global parameter declarations.
*
*-------------------------------------------------------}

CONST   itterations     = 10;    {Maximum model itterations}

        leftscreen      = 1;
        controlscreen   = 2;
        midscreen       = 3;
        modelscreen     = 4;      {Screen and world identifiers}
        line25          = 5;      {for turbo graphix toolbox}
        statusbox       = 6;
        gscrn           = 7;
        hlpscreen       = 8;
        textbox         = 9;

TYPE    timerecord      = RECORD   {System time variable}
                            hrs,
                            min,
                            sec           : INTEGER;
                            dt,
                            itteratetime  : REAL;
                          END;

{-------------------------------------------------------
*
*  The types barrecord is declared in the bargraph unit as follows :
*
*  TYPE barrecord = RECORD
                       xl,yl,x2,y2 : REAL;     - Frame coordinates
                       max,min     : REAL;     - Max and min values
                       len,plc,td  : INTEGER;  - Axis numbering info
                                                 field, decimal places
                                                 and tick mark density.
                       units,title : STRING;   - Axis labels
                       side        : CHAR;     - Title on left,right,
                                                 top or bottom of frame.
                       frame       : BOOLEAN;  - draw frame if true.
```

```
                    END;
*
*
*  The type ptype (parameter type) is declared in the model unit. See
*  the model unit listing for details.
*
*-------------------------------------------------------}


          controlarray = ARRAY[1..4] OF barrecord;
          modelarray   = ARRAY[1..13] OF barrecord;
          midarraytype = ARRAY[1..9] OF barrecord;
          modframetype = (halothane,conductance,bloodflow,volume);
          plotustype   = ARRAY[1..13] OF ptype;
          pmidtype     = ARRAY[1..9] OF ptype;
          pgraphtype   = ARRAY[1..6] OF ptype;

          defaultrecordtype
                       = RECORD
                            l : plotustype;
                            m : pmidtype;
                            g : pgraphtype;
                            xm,ym : REAL;
                            grfl,
                            dpfl  : BOOLEAN;
                            sp    : INTEGER;
                          END;

{-------------------------------------------------------
*
* Define bargraph positions and default parameters for display
*
-------------------------------------------------------}

CONST     controlbar  : controlarray = (
          (x1:15; y1:20; x2:21; y2:90;
  max:4; min:0; len:1; plc:0; td : 4;
  units:'%ATM'; title:'FRESH HALOTHANE';
  side:'r'; frame:false),

           (x1:36; y1:20; x2:42; y2:90;
           max:10; min:0; len:1; plc:0; td : 10;
  units:'L/MIN'; title:'FRESH GAS FLOW';
  side:'r'; frame:false),

          (x1:57; y1:20; x2:63; y2:90;
           max:20; min:0; len:1; plc:0; td : 20;
  units:'/Min'; title:'BREATHING RATE';
  side:'r'; frame:false),

          (x1:78; y1:20; x2:84; y2:90;
  max:1; min:0; len:1; plc:0; td : 10;
  units:'L'; title:'TIDAL VOLUME';
  side:'r'; frame:false));

          modelbar : modelarray = (
          (x1:40; y1:105; x2:60; y2:108;
           max:1; min:0; len:1; plc:0;td : 4;
  units:''; title:'BRAIN WM';
  side:'t'; frame:false),

          (x1:40; y1:95; x2:60; y2:98;
```

```
        max:1; min:0; len:1; plc:0;td : 4;
units:''; title:'BRAIN GM';
side:'t'; frame:false),

       (xl:40; yl:75; x2:60; y2:78;
        max:1; min:0; len:1; plc:0;td : 4;
units:''; title:'LUNG';
side:'t'; frame:false),

       (xl:40; yl:65; x2:60; y2:68;
        max:1; min:0; len:1; plc:0;td : 4;
units:''; title:'HEART M';
side:'t'; frame:false),

       (xl:40; yl:55; x2:60; y2:58;
        max:1; min:0; len:1; plc:0;td : 4;
units:''; title:'RICH TISSUE';
side:'t'; frame:false),

       (xl:40; yl:45; x2:60; y2:48;
        max:1; min:0; len:1; plc:0;td : 4;
units:''; title:'POOR TISSUE';
side:'t'; frame:false),

       (xl:40; yl:35; x2:60; y2:38;
        max:1; min:0; len:1; plc:0;td : 4;
units:''; title:'VISCERA';
side:'t'; frame:false),

       (xl:40; yl:25; x2:60; y2:28;
        max:1; min:0; len:1; plc:0;td : 4;
units:''; title:'MUSCLE';
side:'t'; frame:false),

      (xl:40; yl:15; x2:60; y2:18;
        max:1; min:0; len:1; plc:0;td : 4;
units:''; title:'FAT';
side:'t'; frame:false),

       (xl:40; yl:5; x2:60; y2:8;
        max:1; min:0; len:1; plc:0;td : 4;
units:'???'; title:'???';
side:'t'; frame:false),

       (xl:20; yl:70; x2:25; y2:90;
        max:1; min:0; len:1; plc:0;td : 4;
units:''; title:'VENOUS';
side:'l'; frame:false),

       (xl:30; yl:85; x2:35; y2:95;
        max:1; min:0; len:1; plc:0;td : 4;
units:'BC'; title:'';
side:'l'; frame:false),

       (xl:75; yl:70; x2:80; y2:90;
        max:1; min:0; len:1; plc:0;td : 4;
units:''; title:'ARTERIAL';
side:'r'; frame:false)
       );

       midscreenbar : midarraytype = (
       (xl:10; yl:155; x2:20; y2:205;
```

```
        max:100; min:0; len:1; plc:0;td : 10;
  units:'B/MIN'; title:'HEART RATE';
  side:'r'; frame:false),

        (xl:40; yl:155; x2:50; y2:205;
        max:6; min:0; len:1; plc:0;td : 6;
  units:'L/MIN'; title:'Q OUTPUT';
  side:'r'; frame:false),

        (xl:70; yl:155; x2:80; y2:205;
        max:0.25; min:0; len:2; plc:2;td : 10;
  units:'L'; title:'STROKE VOL';
  side:'r'; frame:false),

        (xl:10; yl:85; x2:20; y2:135;
        max:150; min:0; len:1; plc:0;td : 10;
  units:'mmHg'; title:'MAB PRES';
  side:'r'; frame:false),

        (xl:40; yl:85; x2:50; y2:135;
        max:150; min:0; len:1; plc:0;td : 10;
  units:'mmHg'; title:'SYS BP';
  side:'r'; frame:false),

        (xl:70; yl:85; x2:80; y2:135;
        max:150; min:0; len:1; plc:0;td : 10;
  units:'mmHg'; title:'DIA BP';
  side:'r'; frame:false),

        (xl:10; yl:10; x2:20; y2:65;
        max:100; min:0; len:1; plc:0;  td : 10;
  units:'RATIO'; title:'FA-FI';
  side:'r'; frame:false),

        (xl:40; yl:10; x2:50; y2:65;
        max:10; min:0; len:1; plc:0; td : 10;
  units:'L/MIN'; title:'Q ALV';
  side:'r'; frame:false),

        (xl:70; yl:10; x2:80; y2:65;
        max:4; min:0; len:2; plc:2;  td : 4;
  units:'%ATM'; title:'P INSP';
  side:'r'; frame:false)
        );

{-------------------------------------------------------
*
*                Define the menu strings
*
*-------------------------------------------------------}

        menustring1 : string[80] =
        '1:Help 2:Reset 3:Speed 4:Start/Stop 5:Dump on/off                    Es
        menustring2 : string[80] =
        'COMPARTMENT 1:Concentration  2:Conductance 3:Blood flow  4:Volume      A
        menustring3 : string[80] =
        '1:Print 2:Dump name  3:Dump parameters                               C
        menustring4 : string[80] =
        '1:Select graph parms  2:Graph  3:Clear graph  4:Xaxis  5:Yaxis       S
        exitstring  : string[80] =
        '                        Sure you want to exit? (Y/N)..>
```

```
VAR   dosflag,                  {Return to dos flag.}
      dumpflag,                 {Lotus dump file flag}
      glbgoflag  : BOOLEAN;     {Simulator execute flag}
      glbch      : CHAR;
      dumpfile   : string;      {Lotus Dumpfile name}
      dumpfil    : TEXT;        {Lotus Dumpfile}
      defaultfil : file OF defaultrecordtype; {Default parameter file}
      lotusarray : plotustype; {Parameters sent to lotus dumpfile}
      midarray   : pmidtype;    {Parameters displayed in center window}
      grarray    : pgraphtype; {Parameters displayed by graph window}
      defaultrecord             {Record saved and read from default file}
                 : defaultrecordtype;
      graphgroup,
      grtotal    : INTEGER;     {Graph parameter group (1..4)}
      time       : timerecord; {System time record}
      speed      : INTEGER;     {System speed 1,5,10,15,30,60,600}
      glbframe   : modframetype;{Right window display type}
      glbgraph   : BOOLEAN;     {True if graph window active}
      secondsglb : REAL;        {Total elapsed seconds}
      xmax,ymax  : REAL;        {Graph axes maximums}

{-------------------------------------------------------}

PROCEDURE outputmodel; FORWARD;

{-------------------------------------------------------}

PROCEDURE modelframeinit(frame : modframetype);

{
 This routine draws the model bargraphs in the right hand window
 Possible displays are 1) Halothane concentration, 2) blood flow
 rate, 3) conductance and volume of anaesthetic vapour of each
 compartment of the model.
}

VAR x,y    : REAL;
    i      : INTEGER;
    st     : string;

BEGIN
   FOR i := 1 TO 13 DO modelbar[i].frame := FALSE;
   {
    If modelbar.frame is false then the first call to
    normbar or sidebar draws the frame around the bar-
    graph and labels it. The variable is then set true
    and further calls just draw the bar inside the frame.
   }

   CASE frame OF
      halothane   :BEGIN
                      glbframe := halothane;
                      FOR i := 1 TO 13 DO
                      WITH modelbar[i] DO
                      BEGIN
                         len := 1;
                         plc := 0;
                         IF i < 11 THEN max := 2
                        ELSE max := 4;
                      END;
                    END;
      conductance : BEGIN
```

```
                    glbframe := conductance;
                    FOR i := 1 TO 13 DO
                    WITH modelbar[i] DO
                    BEGIN
                       len := 2;
                       IF i < 11 THEN plc := 2
                      ELSE plc := 0;
                       IF i < 11 THEN max := 0.025
                      ELSE max := 1;
                    END;
                  END;
   bloodflow    :BEGIN
                    glbframe := bloodflow;
                    FOR i := 1 TO 13 DO
                    WITH modelbar[i] DO
                    BEGIN
                       len := 1;
                       plc := 0;
                       IF i <> 12 THEN max := 6
                     ELSE max := 1;
                    END;
                 END;
   volume       :BEGIN
                    glbframe := volume;
                    FOR i := 1 TO 13 DO
                    WITH modelbar[i] DO
                    BEGIN
                       len := 1;
                       plc := 2;
                       CASE i OF
                             1..6,9,11,13 : max := 0.5;
                             7            : max := 2;
                             8            : max := 10;
                             12           : max := 1;
                       END;
                    END;
                 END;
END; {Case}
selectworld(modelscreen);
selectwindow(modelscreen);
setbackground(0);          {Clear the window}
drawborder;

sidebar(modelbar[1],0);
sidebar(modelbar[2],0);
sidebar(modelbar[3],0);
sidebar(modelbar[4],0);
sidebar(modelbar[5],0);    {Draw the frames}
sidebar(modelbar[6],0);
sidebar(modelbar[7],0);
sidebar(modelbar[8],0);
sidebar(modelbar[9],0);
normbar(modelbar[11],0);
normbar(modelbar[12],0);
normbar(modelbar[13],0);

setlinestyle(4);           {Draw the connecting arrows}

{da = draw arrow at da(x,y,direction,size)}

drawline(77.5,107.5,77.5,90);da(60,107.5,'l',6);
drawline(77.5,70,77.5,17.5);da(60,97.5,'l',6);
```

```
   drawline(22.5,107.5,22.5,90);da(75,77.5,'r',6);
   drawline(22.5,70,22.5,17.5);da(60,67.5,'l',6);
   drawline(22.5,107.5,40,107.5);da(60,57.5,'l',6);
   drawline(22.5,97.5,40,97.5);da(60,47.5,'l',6);
   drawline(22.5,67.5,40,67.5);da(60,37.5,'l',6);
   drawline(22.5,57.5,40,57.5);da(60,27.5,'l',6);
   drawline(22.5,47.5,40,47.5);da(60,17.5,'l',6);
   drawline(22.5,37.5,40,37.5);da(22.5,70,'u',6);
   drawline(22.5,27.5,40,27.5);da(22.5,90,'d',6);
   drawline(22.5,17.5,40,17.5);da(55,78,'d',6);
   drawline(60,107.5,77.5,107.5);da(40,77.5,'r',6);
   drawline(60,97.5,77.5,97.5);
   drawline(60,67.5,77.5,67.5);
   drawline(60,57.5,77.5,57.5);
   drawline(60,47.5,77.5,47.5);
   drawline(60,37.5,77.5,37.5);
   drawline(60,27.5,77.5,27.5);
   drawline(60,17.5,77.5,17.5);
   drawline(25,77.5,40,77.5);
   drawline(60,77.5,75,77.5);
   drawline(35,87.5,55,87.5);
   drawline(55,87.5,55,78);
   setlinestyle(0);

   y := 115;                {Label the right hand window}
   CASE frame OF
      halothane   : st := 'Halothane Conc (%Atm)';
      conductance : st := 'Conductance mmHg/L/S';
      bloodflow   : st := 'Blood flow (L/Min)';
      volume      : st := 'Amount Halothane(L)';
   END;
   x := 48-round(length(st)*1.5);
   drawtextw(x,y,2,st);

   outputmodel;             {Fill the bars with data}
END;

{-------------------------------------------------------}

PROCEDURE varframeinit;

{
 This routine initialises the bargraph frames in the middle
 window
}

VAR i : INTEGER;

BEGIN
   selectworld(midscreen);
   selectwindow(midscreen);
   setlinestyle(0);
   drawborder;
   FOR i := 1 TO 9 DO
   BEGIN
      midscreenbar[i].frame := FALSE;
      normbar(midscreenbar[i],0);
   END;
END;

{-------------------------------------------------------}
```

```
PROCEDURE controlframeinit;

{
 This routine initialises the control window bargraphs
}

BEGIN
   setlinestyle(0);
   selectworld(controlscreen);
   selectwindow(controlscreen);
   GOTOXY(7,14);
   WRITE('CONTROLS');
   normbar(controlbar[1],0);
   normbar(controlbar[2],0);
   normbar(controlbar[3],0);
   normbar(controlbar[4],0);
END;

{-------------------------------------------------------}

PROCEDURE initgraphscreen;

{
 This routine defines the graphic screen window coordinates on
 the screen, and defines the world coordinates for each window.
 It then calls the routines to initialise each window.
}

VAR i      : INTEGER;

BEGIN
   defineworld(controlscreen,0,0,100,120);
   defineworld(modelscreen,12,10,88,120);
   defineworld(midscreen,-5,-5,90,215);
   defineworld(gscrn,0,0,xmax,ymax);
   definewindow(leftscreen,0,0,xmaxglb DIV 4,ymaxglb DIV 2);
   definewindow(controlscreen,0,(ymaxglb DIV 2 + 1),xmaxglb DIV 4,ymaxglb-15);
   definewindow(midscreen,(xmaxglb div 4)+1,0,3*(xmaxglb div 5),ymaxglb-15);
   definewindow(modelscreen,3*(xmaxglb div 5)+1,0,xmaxglb,ymaxglb-15);
   definewindow(gscrn,(xmaxglb DIV 4)+2,20,xmaxglb-2,ymaxglb-75);
   definewindow(hlpscreen,0,0,xmaxglb,ymaxglb);
   definetextwindow(statusbox,4,2,18,12,4);
   definetextwindow(textbox,23,14,48,22,4);
   definewindow(16,0,0,xmaxglb,ymaxglb);

   selectworld(controlscreen);
   setlinestyle(0);
   FOR i := 1 TO 4 DO
   BEGIN
      selectwindow(i);
      drawborder;
   END;

   controlframeinit;
   modelframeinit(halothane);
   varframeinit;

   gotoxy(5,2);
   WRITELN('UCTAS Ver 2.0');
   selectwindow(statusbox);
   drawborder;
END;
```

```
{-----------------------------------------------------}

PROCEDURE checkcommandline;

{
 This routine checks for the special keys control, shift and alt.
 The corresponding menu string is updated on the bottom line of
 the graphics screen if one is pushed.
}

VAR keyboard,
    i         : INTEGER;
    str       : string[80];

BEGIN
   keyboard := kbdstatus;
   CASE keyboard AND $0F OF
      1,2 :str := menustring4;
      4   :str := menustring3;
      8   :str := menustring2;
      ELSE str := menustring1;
   END; {Case}
   GOTOXY(1,25);
   FOR i := 1 TO length(str) DO
   BEGIN
      gotoxy(i,25);
      IF str[i] IN ['0'..'9'] THEN
      BEGIN
         invglb := TRUE;
         WRITE(str[i]);
         invglb := FALSE;
      END
     ELSE
      WRITE(str[i]);
   END;
END;

{-----------------------------------------------------}

PROCEDURE outtext;

{
 This routine writes the value of the control variables above
 the corresponding control bargraph.
}

BEGIN
   WITH modelparameters DO
   BEGIN
      invglb := TRUE;
      GOTOXY(3,15);
      WRITE(pfr:1:1);
      GOTOXY(7,15);
      IF Qfr < 10 THEN WRITE(Qfr:1:1)
     ELSE
      WRITE(Qfr:1:0);
      GOTOXY(11,15);
      WRITE(BR:1:1);
      GOTOXY(16,15);
      WRITE(Vtidal:1:3);
      invglb := FALSE;
```

```
   END;
END;

{-------------------------------------------------------}

PROCEDURE outputcontrol(delta : INTEGER);

{
 This routine updates the control variable bargraph specified by
 delta. 1 = left hand control bar to 4 = right hand bar.
}

BEGIN
   WITH modelparameters DO
   BEGIN
      selectworld(controlscreen);
      selectwindow(controlscreen);
      CASE delta OF
         1 : normbar(controlbar[1],Pfr);
         2 : normbar(controlbar[2],Qfr);
         3 : normbar(controlbar[3],BR);
         4 : normbar(controlbar[4],Vtidal);
      END;
      outtext;
   END;
END;

{-------------------------------------------------------}

PROCEDURE indicatecontrol(delta : INTEGER);

{
 This routine draws a small square underneath the control
 bargraph specified by delta.
}

TYPE bararray = ARRAY[1..4] OF barrecord;

CONST bar : bararray = (
            (x1:15; y1:1; x2:21; y2:5),
            (x1:36; y1:1; x2:42; y2:5),
            (x1:57; y1:1; x2:63; y2:5),
            (x1:78; y1:1; x2:84; y2:5));

VAR i : INTEGER;

{.....................................................}

PROCEDURE drawbar(bargraph  : barrecord;
                  value,xmax : REAL);
{
 Routine to draw square
}

VAR yy,yy1,yy2 : INTEGER;
    perc       : REAL;

BEGIN
   WITH bargraph DO
   BEGIN
      perc := (value/xmax)*100;
      IF perc > 100 THEN perc := 100;
```

```
      IF perc < 0 THEN perc := 0;
      yy := windowy(y1+(y2-y1)*perc/100);
      IF value > 0 THEN
         drawsquarec(windowx(x1),windowy(y1),windowx(x2),yy,true);
      setcolorblack;
      IF value = 0 THEN yy1 := yy ELSE yy1 := yy - 1;
      yy2 := windowy(y2);
      IF yy1 > yy2 THEN
         drawsquarec(windowx(x1),yy1,windowx(x2),yy2,true);
      setcolorwhite;
   END;
END;

{.....................................................}

BEGIN
   selectworld(controlscreen);
   selectwindow(controlscreen);
   FOR i := 1 TO 4 DO
      drawbar(bar[i],0,100);

   CASE delta OF
      1 :drawbar(bar[1],100,100);
      2 :drawbar(bar[2],100,100);
      3 :drawbar(bar[3],100,100);
      4 :drawbar(bar[4],100,100);
   END;
END;

{-------------------------------------------------------}

PROCEDURE outputmodel;

{
 This routine updates all the bargraphs in the right hand window
}

BEGIN
   selectworld(modelscreen);
   selectwindow(modelscreen);
   WITH modelparameters DO
   CASE glbframe OF        {Check what parameters are being plotted}
      halothane   :BEGIN
                      sidebar(modelbar[1],Pw);  {,'Brain WM',true);}
                      sidebar(modelbar[2],Pg);  {,'Brain GM',true);}
                      sidebar(modelbar[3],Pl);  {,'Lung',true);}
                      sidebar(modelbar[4],Ph);  {,'Heart M',true);}
                      sidebar(modelbar[5],Pr);  {,'Rich tissue',true);}
                      sidebar(modelbar[6],Pp);  {,'Poor tissue',true);}
                      sidebar(modelbar[7],Pi);  {,'Viscera',true);}
                      sidebar(modelbar[8],Pm);  {,'Muscle',true);}
                      sidebar(modelbar[9],Pf);  {,'Fat',true);}

                      normbar(modelbar[11],Pv); {,'Venous',true);}
                      normbar(modelbar[12],Pbc); {,'BC',true);}
                      normbar(modelbar[13],Pa); {,'Arterial',true);}
                   END;
      conductance : BEGIN
                       sidebar(modelbar[1],Gw);  {,'Brain WM',true);}
                       sidebar(modelbar[2],Gg);  {,'Brain GM',true);}
                       sidebar(modelbar[3],0);  {,'Lung',true);}
                       sidebar(modelbar[4],Gh);  {,'Heart M',true);}
```

```
                        sidebar(modelbar[5],Gr);  {,'Rich tissue',true);}
                        sidebar(modelbar[6],Gp);  {,'Poor tissue',true);}
                        sidebar(modelbar[7],Gi);  {,'Viscera',true);}
                        sidebar(modelbar[8],Gm);  {,'Muscle',true);}
                        sidebar(modelbar[9],Gf);  {,'Fat',true);}

                     END;
      bloodflow    :BEGIN
                        sidebar(modelbar[1],Qw);  {,'Brain WM',true);}
                        sidebar(modelbar[2],Qg);  {,'Brain GM',true);}
                        sidebar(modelbar[3],Ql);  {,'Lung',true);}
                        sidebar(modelbar[4],Qh);  {,'Heart M',true);}
                        sidebar(modelbar[5],Qr);  {,'Rich tissue',true);}
                        sidebar(modelbar[6],Qp);  {,'Poor tissue',true);}
                        sidebar(modelbar[7],Qi);  {,'Viscera',true);}
                        sidebar(modelbar[8],Qm);  {,'Muscle',true);}
                        sidebar(modelbar[9],Qf);  {,'Fat',true);}

                        normbar(modelbar[11],Q); {,'Venous',true);}
                        normbar(modelbar[13],Q); {,'Arterial',true);}
                     END;
      volume       :BEGIN
                        sidebar(modelbar[1],Aw);  {,'Brain WM',true);}
                        sidebar(modelbar[2],Ag);  {,'Brain GM',true);}
                        sidebar(modelbar[3],Al);  {,'Lung',true);}
                        sidebar(modelbar[4],Ah);  {,'Heart M',true);}
                        sidebar(modelbar[5],Ar);  {,'Rich tissue',true);}
                        sidebar(modelbar[6],Ap);  {,'Poor tissue',true);}
                        sidebar(modelbar[7],Ai);  {,'Viscera',true);}
                        sidebar(modelbar[8],Am);  {,'Muscle',true);}
                        sidebar(modelbar[9],Af);  {,'Fat',true);}

                        normbar(modelbar[11],Aa); {,'Venous',true);}
                        normbar(modelbar[12],Abc);
                        normbar(modelbar[13],Av); {,'Arterial',true);}
                     END;
   END; {Case}
END;

{-------------------------------------------------------}

PROCEDURE outputvariable;

{
 This routine updates the bargraphs in the middle window
}

VAR i : INTEGER;

BEGIN
   selectworld(midscreen);
   selectwindow(midscreen);
   FOR i := 1 TO 9 DO
     normbar(midscreenbar[i],getp(midarray[i]));
END;

{-------------------------------------------------------}

PROCEDURE outtime;

{
 This routine updates the time in the status window
```

```
}

VAR hs,ms,ss : STRING[2];
    st       : STRING;

BEGIN
   WITH time DO
   BEGIN
      hms(secondsglb/3600,hrs,min,sec);  {Convert real to 3 hms integers}
      IF hrs >= 24 THEN
      BEGIN
         hrs := 0;
         min := 0;
         sec := 0;
         secondsglb := 0;
      END;
      str(hrs,hs); IF hrs < 10 THEN hs := '0'+hs;
      str(min,ms); IF min < 10 THEN ms := '0'+ms;  {Fill in blanks}
      str(sec,ss); IF sec < 10 THEN ss := '0'+ss;
      st := hs+':'+ms+':'+ss;                      {Convert to string}
      gotoxy(4,4);
      WRITELN('Time = ',st);

     {additions include memory available and dumpfile name}

      gotoxy(4,10);
      WRITELN('Mem = ',memavail);
      IF dumpflag then st:=DUMPFILE ELSE st:='NONE     ';
      GOTOXY(4,12);
      WRITELN(st);
   END;
END;

{-------------------------------------------------------}

PROCEDURE outspeed;

{
This routine updates the current speed in the status window
}

VAR st : string;

BEGIN
   gotoxy(4,6);
   str(speed,st);
   IF speed < 10 THEN st := ' '+st;
   WRITELN('Speed = ',st+'X ');
END;

{-------------------------------------------------------}

PROCEDURE outstatus;

{
Update the simulator status in the status window. The flag
is changed when F4 is pressed. See PROCEDURE simulator further
on.
}

BEGIN
   gotoxy(4,8);
```

```
   CASE glbgoflag OF
      true : WRITELN('Status = Go  ');
      false :WRITELN('Status = Stop');
   END;
END;

{-------------------------------------------------------}

PROCEDURE outputgraph;

{
 Update the graph in the graph window.
 Routine plotpoints comes from UNIT axes.
 Plotpoints(x,y,linenumber,linestyle)
}

VAR i : INTEGER;
    r : REAL;       {Temporary real for plotting}

BEGIN
   selectworld(gscrn);
   selectwindow(gscrn);
   FOR i := 1 TO 6 DO
   BEGIN
      r := getp(grarray[i]);    {Get the value}
      IF r <> -99 THEN          {Check if it should be plotted}
      plotpoints(secondsglb/60,r,i-1,0);
   END;
END;

{-------------------------------------------------------}

PROCEDURE outputlotus;

{
Send parameters to the lotus dump file as text line. Order is
Time 1stparameter 2ndparameter ........ lastparameter.
A space is left between each one.
The parameter names are written on the first line of the file
when a new file is opened (see below).
}

VAR i : INTEGER;

BEGIN
   WRITE(dumpfil,secondsglb:10);       {First one is time}
   FOR i := 1 TO 13 DO                 {Check if data should be sent}
      IF lotusarray[i] <> p_null THEN
         WRITE(dumpfil,getp(lotusarray[i]):10:6,' ');
   WRITELN(dumpfil);                   {New line after all sent}
END;

{-------------------------------------------------------}

PROCEDURE initdumpfile;

{
Initialise the lotus dump file and send names of selected
output variables to it
}

VAR i : INTEGER;
```

```
    b : barrecord;

BEGIN
   ASSIGN(dumpfil,dumpfile);      {open dumpfile WRITE only}
   REWRITE(dumpfil);              {Old file with same name will be}
   WRITE(dumpfil,'"      TIME"');{destroyed}

   FOR i := 1 TO 13 DO
   IF lotusarray[i] <> p_null THEN
   BEGIN
      initbarrecord(lotusarray[i],b);    {Send names in between inverted}
      WRITE(dumpfil,'"',b.title:10,'"'); {commas all on same line}
   END;

   WRITELN(dumpfil);                     {New line}
   outputlotus;                          {Send initial data on next line}
END;

{-------------------------------------------------------}

FUNCTION exist(fname : string) : BOOLEAN;

{
Test if a file exists. Returns TRUE if it does ELSE returns FALSE.
}

VAR fil  : text;
    temp : word;

BEGIN
    ASSIGN(fil,fname);
   {$I-}
    RESET(fil);                   {Try and open the file}
  {$I+}
    temp := ioresult;
    exist := (temp = 0);          {If no error then it exists}
    IF temp = 0 THEN CLOSE(fil); {Close it if it was opened}
END;

{-------------------------------------------------------}

PROCEDURE getdumpfilename;

{
Prompt for a new file name. Close the old file. Initialise the new
file. Called from simulator procedure with CNTROL F2.
}

VAR roger : BOOLEAN;
    ch    : CHAR;

BEGIN
   clrscr;
   center(1,'DUMP FILENAME ROUTINE');
   puts(5,5,'File '+dumpfile+' closed');
   roger := false;
   REPEAT
      puts(5,7,'Enter new file name ..>');
      READLN(dumpfile);                    {Get new name}
      IF length(dumpfile) > 0 THEN
      BEGIN
         IF exist(dumpfile) THEN
```

```
         BEGIN
            center(24,'Error: File already exists. Overwrite (y/n)');
            WRITE(^G);
            ch := readkey;                    {Check if OK to destroy}
            IF upcase(ch) = 'Y' THEN          {existing file}
            BEGIN
               close(dumpfil);
               initdumpfile;                  {If so continue else prompt}
               roger := true;                 {for new name again}
            END;
         END
        ELSE
         BEGIN
            close(dumpfil);                   {Close old file}
            initdumpfile;                     {Initialise new one}
            roger := true;
         END;
      END
     ELSE
      roger := TRUE;
   UNTIL roger;                               {Exit to calling routine}
END;

{-------------------------------------------------------}

PROCEDURE getdumpfileparameters;

{
Select 13 parameters to be sent to the lotus dump file.
Called by CNTRL F3 from simulator procedure.
}

VAR pstart,pend,
    pcnt          : ptype;
    x,y,yy        : INTEGER;
    esc,esc2      : BOOLEAN;
    dumppos,i,
    choice        : INTEGER;
    b             : barrecord;
    bstring       : string[15];
    st            : string;

BEGIN
   clrscr;
   message(1,'SELECT DUMPFILE PARAMETERS');
   st := '';
   for i := 1 TO 50 DO st := st + chr(205); {String = line of dashes}
   x := 3;
   yy := 9;                          {Set start x,y coords}
   pstart := p_BR;                   {Set start and end parameters that}
   pend := p_Av;                     {May be selected}
   FOR pcnt := pstart TO pend DO     {Initialise the screen first}
   BEGIN
      gotoxy(x,yy);
      initbarrecord(pcnt,b);         {Get info about the parameter into b}
      WRITE(b.title);                {Write its name on the screen}
      x := x + 20;
      IF x > 63 THEN
      BEGIN
         x := 3;                     {Update the x and y coords where the}
         yy := yy + 1;               {next title will be written}
      END;
```

```
    END;
    drawbox(1,8,79,25);              {Draw boxes around the data}
    drawbox(1,2,79,7);

    dumppos := 1;                    {Start at first dump parameter}
    REPEAT
       x := 3;
       y := 3;
       bstring := '                ';
       FOR i := 1 TO 13 DO           {Write all the default or current}
       BEGIN                         {choices in top part of screen}
          gotoxy(x,y);
          initbarrecord(lotusarray[i],b);
          WRITE(b.title);
          WRITE(copy(bstring,1,15-length(b.title))); {Clear end of string}
          x := x + 20;
          IF x > 63 THEN
          BEGIN
             x := 3;
             y := y + 1;
          END;
       END;
       center(7,'Select with cursor and Return. Esc when finished');
       center(25,st);
       selectwithbar(3,3,4,20,15,13,dumppos,esc); {Move around with cursor}
       IF NOT(esc) THEN                           {keys and hit return when}
       BEGIN                                      {choice is made}
          choice := 1;                            {Move to bottom list and }
          center(7,st);                           {select with cursor keys }

          center(25,'Select parameter with cursor. Esc for nothing');
          selectwithbar(3,9,4,20,15,ord(pend)-ord(pstart)+1,choice,esc2);

          IF esc2 THEN lotusarray[dumppos] := p_null {If esc hit then set dump}
        ELSE                                         {value = to null. ie no }
          BEGIN                                      {parameter will be sent to}
             pcnt := pred(pstart);                   {the dump file for this one}
             FOR i := 1 TO choice DO pcnt := succ(pcnt);
             lotusarray[dumppos] := pcnt;            {Update new dump choice in}
          END;                                       {the lotusarray}
       END;
    UNTIL esc;
END;

{-------------------------------------------------------}

PROCEDURE outputresults;

{
This procedure called by the simulator routine updates all data.
Updates the graph if graph screen is active, else updates the
bargraphs. Always updates the time and sends a line of data to
the dump file if dumping is active.
}

BEGIN
   IF glbgraph THEN outputgraph
  ELSE
   BEGIN
      outputmodel;
      outputvariable;
   END;
```

```
   outtime;
   IF dumpflag THEN outputlotus;
END;

{-------------------------------------------------------}

PROCEDURE resetmodel;

{
Reset the model variables to there default unanaesthetised state.
Called from the simulator procedure with F2.
}

VAR i : INTEGER;

BEGIN
   initialisemodelstart;            {Reset the model}

   FOR i := 1 TO 6 DO
   BEGIN
      ox[i-1] := 0;                 {Reset the graph}
      oy[i-1] := getp(grarray[i]);
      oxyinit[i-1] := TRUE;
      linecount[i-1] := i;
   END;
   maxcount := round(xmax*6/speed);
   IF maxcount < 0 THEN maxcount := 1;
   pointsize := 2;

   secondsglb := 0;                 {Reset the time}
   WITH time DO
   BEGIN
      hrs := 0;
      min := 0;
      sec := 0;
      dt  := speed*(1/(itterations*60));
      itteratetime := 60*itterations*dt;
   END;
END;

{-------------------------------------------------------}

PROCEDURE idgraphaxes;

{
Identify the graph parameters being plotted in the graph window.
The names of the parameters being plotted are stored in the array
lineid. This is used later by the routine idgraph which exists in
the UNIT axes.
}

VAR i : INTEGER;
    b : barrecord;

BEGIN
   FOR i := 1 TO 6 DO
   BEGIN
      initbarrecord(grarray[i],b); {Get the info about the parameter}
      lineid[i-1] := b.title;
   END;
END;
```

```
{-------------------------------------------------------}

PROCEDURE drawgraphframe;

{
Initialise the graph window. Draw the frame and initialise the
axes titles.

Axes procedure called as follows :

PROCEDURE drawaxes(win,wrld              : INTEGER;
                   xtick,ytick           : REAL;
                   xaxis,yaxis,
                   title                 : string;
                   xlen,xpl,ylen,ypl     : INTEGER;
                   id                    : BOOLEAN);

Parameters are :
        win   - Predeclared window to draw fraph in.
        wrld  - Predeclared world (ie 0,0 to 10,100).
        xtick,
        ytick - Tick density along x and y axis.
        xaxis,
        yaxis,
        title - Axis titles and main title of graph.
        xlen,
        xpl,
        ylen,
        ypl   - Field length and dec places of axis numbers.

NOTE!!! drawaxes redefines the world wrld to fit the axis and
        lables into the window. However you can still plot
        within the graph using the original world as the old
        world lower left coords are now at the x and y axis
        junction.
}

VAR ytick,
    xtick     : REAL;
    ygraphstr,
    titlegrstr : string;
    xlen,ylen,
    xpl,ypl    : INTEGER;

BEGIN
   IF ymax < 1 THEN ypl := 2 ELSE  {Set y decimal places}
   IF ymax < 10 THEN ypl := 1 ELSE
   ypl := 0;
   ylen := 1;
   ytick := ymax/8;                {Set axis tick densities}
   xtick := xmax/4;
   CASE graphgroup OF              {Set axis lables}
      1 : BEGIN
             ygraphstr := '% ATMOSPHERE';
             titlegrstr := 'HALOTHANE PARTIAL PRESSURE';
          END;
      2 : BEGIN
             ygraphstr := 'LITRES/MIN';
             titlegrstr := 'COMPARTMENT BLOOD FLOW';
          END;
      3 : BEGIN
             ygraphstr := 'LITRES PER MIN PER mmHg';
```

```
            titlegrstr := 'COMPARTMENT CONDUCTANCE';
         END;
      4 : BEGIN
            ygraphstr := 'LITRES HALOTHANE';
            titlegrstr := 'COMPARTMENT AMOUNTS';
         END;
      5 : BEGIN
            ygraphstr := '?';
            titlegrstr := 'MIXED PARAMETERS';
         END;
   END;

   drawaxes(gscrn,gscrn,xtick,ytick,
            'TIME (Minutes)',ygraphstr,titlegrstr,
            2,2,ylen,ypl,true);

   idgraphaxes;                        {Id the parameters}
   idgraph(gscrn,6,1);

END;

{-------------------------------------------------------}

PROCEDURE getgraphparms;

{
A procedure to select the parameters to be graphed. Works in
the same way as select dump file parameters. The only difference
is that one of 5 groups of parameters may be selected first. This
is so that parameters with similar values may be graphed together
without confusing y axes scales.
}

VAR esc,
    esc2,
    exit    : BOOLEAN;
    ch      : CHAR;
    b       : barrecord;
    st      : string;
    bstring : string[15];
    i,x,y,
    oldgroup,
    graphpos,
    choice : INTEGER;
    pstart,
    pend,
    pcnt    : ptype;

BEGIN
   st := '';
   for i := 20 TO 69 DO st := st + chr(205);
   bstring := '               ';
   clrscr;
   message(1,'SELECT GRAPH PARAMETERS');
   message(3,'You may select up to 6 parameters at a time to display on the graph')
   message(5,'All 6 parameters must be chosen from 1 of the following 5 groups');

   message(7,'HALOTHANE CONCENTRATION');
   message(8,'COMPARTMENT BLOOD FLOW ');
   message(9,'COMPARTMENT CONDUCTANCE');
   message(10,'COMPARTMENT AMOUNTS    ');
   message(11,'ALL BUT AMOUNTS        ');
```

```
message(25,'Select a group using the cursor keys. Esc to leave unchanged');
oldgroup := graphgroup;
selectwithbar(28,7,1,25,25,5,graphgroup,esc);

IF NOT(esc) THEN
BEGIN
   clrscr;
   IF oldgroup <> graphgroup THEN
   FOR i := 1 TO 6 DO grarray[i] := p_null;
   message(1,'CURRENT GRAPH PARAMETERS');

   CASE graphgroup OF
      3 : BEGIN pstart := p_Gf; pend := p_Gw; END;
      1 : BEGIN pstart := p_Pa; pend := p_Pw; END;
      2 : BEGIN pstart := p_Qf; pend := p_Qw; END;
      5 : BEGIN pstart := p_BR; Pend := p_Qw; END;
      4 : BEGIN pstart := p_Abc; Pend := p_Av; END;
   END;

   x := 8;
   y := 7;
   FOR pcnt := pstart TO pend DO
   BEGIN
      gotoxy(x,y);
      initbarrecord(pcnt,b);
      WRITE(b.title);
      x := x + 25;
      IF x > 60 THEN
      BEGIN
         x := 8;
         y := y + 1;
      END;
   END;
   drawbox(1,6,79,y+1);
   drawbox(1,2,79,5);

   graphpos := 1;
   REPEAT
      FOR i := 1 TO 2 DO
      BEGIN
         gotoxy(8,2+i);
         initbarrecord(grarray[3*i-2],b);
         WRITE(b.title);
         WRITE(copy(bstring,1,15-length(b.title)));
         gotoxy(33,2+i);
         initbarrecord(grarray[3*i-1],b);
         WRITE(b.title);
         WRITE(copy(bstring,1,15-length(b.title)));
         gotoxy(58,2+i);
         initbarrecord(grarray[3*i],b);
         WRITE(b.title);
         WRITE(copy(bstring,1,15-length(b.title)));
      END;
      center(5,' Select with cursor and Return. Esc when finished ');
      center(y+1,st);
      selectwithbar(8,3,3,25,15,6,graphpos,esc);
      IF NOT(esc) THEN
      BEGIN
         choice := 1;
         center(5,st);
         center(y+1,'Select parameter with cursor. Esc for nothing');
         selectwithbar(8,7,3,25,15,ord(pend)-ord(pstart)+1,choice,esc2);
```

```
            IF esc2 THEN grarray[graphpos] := p_null
           ELSE
            BEGIN
               pcnt := pred(pstart);
               FOR i := 1 TO choice DO pcnt := succ(pcnt);
               grarray[graphpos] := pcnt;
            END;
         END;
      UNTIL esc;
   END;
END;

{-------------------------------------------------------}

PROCEDURE getinteger(VAR ym : REAL; esc : BOOLEAN);

{
A procedure to read an integer number on the lower line of the
screen while in graphics mode.
}

VAR cr   : BOOLEAN;
    str  : string;
    ch   : CHAR;
    k,
    xpos : INTEGER;
    temp : REAL;

BEGIN
   xpos := 38;
   esc := FALSE;
   cr := FALSE;
   str := '';
   REPEAT
      ch := readkey;
      CASE ch OF
         #27 : esc := true;
         #13 : cr := true;
'0'..'9','.' : str := str + ch;
         #8  : BEGIN
                  gotoxy(xpos+length(str)-1,25);
                  WRITELN(' ');
                  str := copy(str,1,length(str)-1);
               END;
         ELSE beep(1000);
      END;
      GOTOXY(38,25);
      WRITELN(str);
   UNTIL cr OR esc;
   IF NOT(esc) THEN
   BEGIN
      val(str,temp,k);
      IF k = 0 THEN ym := temp
     ELSE
      BEGIN
         beep(1000);
         esc := TRUE;
      END;
   END;
END;

{-------------------------------------------------------}
```

```
PROCEDURE itteratemodel;

{
A procedure to iterate the model equations 10 times over a
time period of itteratetime. A star in the status screen is
written while the iteration takes place.
}

VAR i : INTEGER;

BEGIN
   GOTOXY(18,8); WRITELN('*');
   FOR i := 1 TO itterations DO model(time.dt);
   secondsglb := secondsglb + time.itteratetime;
   GOTOXY(18,8); WRITELN(' ');
END;

{-------------------------------------------------------}
```

```
{-------------------------------------------------------}

PROCEDURE simulator;

{
The main subroutine. It primary Functions are as follows :

1/ initialise all the screens when called.
2/ get commands from the keyboard and execute them.
3/ Run the simulator model in a loop and update the graphics

}

CONST Pchange = 0.1;
      Brchange = 0.1;
      vtidalchange = 0.025;
      Qchange = 0.1;

VAR dosflag,
    esc,temp : BOOLEAN;
    ch       : CHAR;
    keyboard : INTEGER;
    i        : INTEGER;
    delta    : INTEGER;   {Active control variable}
                          {1 = Pins}
                          {2 = BR  }
                          {3 = Vtidal}

BEGIN
   initgraphic;               {initialise graphics mode}
   setbreakoff;
   GOTOXY(34,12);
   WRITELN('Please  wait');
   GOTOXY(34,14);             {Initialise graphics screen}
   WRITELN('Initialising');
   selectscreen(2);
   initgraphscreen;

   glbgoflag := false;  {Dont start iterating until F4 pushed}

   outstatus;
   temp := glbgraph;
   glbgraph := FALSE;
   outputresults;
   glbgraph := temp;
   outspeed;
   FOR delta := 1 TO 4 DO outputcontrol(delta);
   delta := 1;
   indicatecontrol(delta);

   storewindow(16);
   drawgraphframe;              {Initialise graph window}
   storewindow(gscrn);
   restorewindow(-16,0,0);
   selectscreen(1);
   swapscreen;
   IF glbgraph THEN
   BEGIN
      storewindow(16);
      restorewindow(-gscrn,0,0);
   END;
```

```
dosflag := FALSE;
REPEAT
```

```
REPEAT
   IF glbgoflag THEN
   BEGIN
      itteratemodel;
      copyscreen;              {If glbgoflag then iterate model}
      selectscreen(2);         {update graphics in RAM window and}
      outputresults;           {copy result onto screen}
      swapscreen;
      IF NOT(glbgraph) THEN
      BEGIN
         restorewindow(-gscrn,0,0);
         outputgraph;          {Update the graph in the RAM window}
                               {If its not active}
         storewindow(gscrn);
      END;
      selectscreen(1);
   END;
   checkcommandline;           {Check for special keys such as shift}
                               {control and alt}
UNTIL keypressed;              {If a key is pressed then check if}
ch := readkey;                 {it is a command}
IF keypressed THEN
BEGIN                          {Check for function and cursor keys}
   ch := readkey;
   CASE ord(ch) OF
   F1:BEGIN                                {Help}
         storewindow(hlpscreen);
         leavegraphic;
         help('uctas.hlp');
         entergraphic;
         restorewindow(-hlpscreen,0,0);
      END;
   F2: BEGIN                               {Reset}
          glbgoflag := FALSE;
          resetmodel;
          IF NOT(glbgraph) THEN outputresults;
          FOR delta := 1 TO 4 DO outputcontrol(delta);
          delta := 1;
          indicatecontrol(delta);
          outstatus;
          outtime;
       END;
   F3: BEGIN                                {Change speed}
          IF speed = 1 THEN speed := 5
          ELSE
           IF speed = 5 THEN speed := 10
          ELSE
           IF speed = 10 THEN speed := 15
          ELSE
           IF speed = 15 THEN speed := 30
          ELSE
           IF speed = 30 THEN speed := 60
          ELSE
           IF speed = 60 THEN speed := 600
          ELSE
           IF speed = 600 THEN speed := 1;

           maxcount := round(xmax*6/speed);
           IF maxcount < 0 THEN maxcount := 1;

           WITH time DO
           BEGIN
```

```
            dt  := speed*(1/(itterations*60));
            itteratetime := 60*itterations*dt;
         END;
        outspeed;
      END;
F4:  BEGIN
        glbgoflag := NOT(glbgoflag);   {Start/stop}
        outstatus;
     END;
F5:  BEGIN
        dumpflag:=NOT(dumpflag);       {Toggle dump file on/off}
        outtime;
     END;
F6:;
F7:;
F8:;     {All unused and available for future commands}
F9:;
F10:;
AF1: IF NOT(glbgraph) THEN     {Select halothane conc in }
     BEGIN                     {right hand screen}
        copyscreen;
        selectscreen(2);
        modelframeinit(halothane);
        selectscreen(1);
        swapscreen;
     END;
AF2: IF NOT(glbgraph) THEN     {select conductance}
     BEGIN
        copyscreen;
        selectscreen(2);
        modelframeinit(conductance);
        selectscreen(1);
        swapscreen;
     END;
AF3: IF NOT(glbgraph) THEN     {Select blood flow}
     BEGIN
        copyscreen;
        selectscreen(2);
        modelframeinit(bloodflow);
        selectscreen(1);
        swapscreen;
     END;
AF4:IF NOT(glbgraph) THEN      {Select amounts or volume}
     BEGIN
        copyscreen;
        selectscreen(2);
        modelframeinit(volume);
        selectscreen(1);
        swapscreen;
     END;
AF5:;
AF6:;
AF7:;    {All unused and available for future commands}
AF8:;
AF9:;
AF10:;
SF1:BEGIN                          {Get graph parameters}
      storewindow(hlpscreen);
      leavegraphic;
      getgraphparms;
      entergraphic;
      restorewindow(-hlpscreen,0,0);
```

```
      CASE graphgroup OF
         1 : ymax := 4;
         2 : ymax := 6;
         3 : ymax := 0.02;          {Set up y axis}
         4 : ymax := 2;
         5 : ymax := 10;
      END;
      defineworld(gscrn,0,0,xmax,ymax);
      axesdrawn[gscrn] := FALSE;
      selectscreen(2);
      drawgraphframe;               {Redraw graph and id axes}
      FOR i := 1 TO 6 DO
      BEGIN
         oy[i-1] := getp(grarray[i]);
         linecount[i-1] := i;
      END;
      storewindow(gscrn);
      selectscreen(1);
      IF glbgraph THEN restorewindow(-gscrn,0,0);
   END;
SF2:BEGIN                     {Select/deselect graph window}
      IF glbgraph THEN
      BEGIN
         storewindow(gscrn);
         restorewindow(-16,0,0);
         outstatus;
         for delta := 1 TO 4 DO
            outputcontrol(delta);
         delta := 1;
         indicatecontrol(delta);
      END
      ELSE
      BEGIN
         storewindow(16);
         restorewindow(-gscrn,0,0);
      END;
      glbgraph := NOT(glbgraph);
      IF NOT(glbgraph) THEN outputresults;
    END;
SF3:BEGIN                       {Clear graph by redrawing it}
      selectscreen(2);
      drawgraphframe;
      storewindow(gscrn);
      selectscreen(1);
      IF glbgraph THEN restorewindow(-gscrn,0,0);
   END;
SF4:BEGIN                       {Enter new time scale on x axis}
      gotoxy(1,25);
      WRITELN('Enter new graph time axis maximum  ..>
      getinteger(xmax,esc);
      BEGIN
         defineworld(gscrn,0,0,xmax,ymax);
         axesdrawn[gscrn] := FALSE;
         selectscreen(2);
         drawgraphframe;
         storewindow(gscrn);
         selectscreen(1);
         IF glbgraph THEN restorewindow(-gscrn,0,0);
         maxcount := round(xmax*6/speed);
         IF maxcount < 0 THEN maxcount := 1;
      END;
    END;
```

```
SF5:BEGIN                          {Change the y axis maximum}
       gotoxy(1,25);
       WRITELN('Enter new graph y axis maximum  ..>
       getinteger(ymax,esc);
       BEGIN
          defineworld(gscrn,0,0,xmax,ymax);
          axesdrawn[gscrn] := FALSE;
          selectscreen(2);
          drawgraphframe;
          storewindow(gscrn);
          selectscreen(1);
          IF glbgraph THEN restorewindow(-gscrn,0,0);
       END;
    END;
SF6:;
SF7:;    {All unused and available for future commands}
SF8:;
SF9:;
SF10:;
CF1 :hardcopy(false,1);          {Print the screen on an Epson}
CF2 :BEGIN                       {Select new dumpfile and close}
        storewindow(hlpscreen);{the old one}
        leavegraphic;
        getdumpfilename;
        entergraphic;
        restorewindow(-hlpscreen,0,0);
     END;
CF3:BEGIN                        {Get dumpfile parameters}
        storewindow(hlpscreen);
        leavegraphic;
        getdumpfileparameters;
        entergraphic;
        restorewindow(-hlpscreen,0,0);
     END;
CF4:;
CF5:;
CF6:;
CF7:;    {All unused and available for future commands}
CF8:;
CF9:;
CF10:;
pgup    :  BEGIN    {Increase the value of the current control}
                    {variable by big amount}
             WITH modelparameters DO
             BEGIN
                CASE delta OF
                   1 :Pfr := Pfr + 5*Pchange;
                   2 :Qfr := Qfr + 5*qchange;
                   3 :BR := BR + 10*Brchange;
                   4 :Vtidal := Vtidal + 5*vtidalchange;
                END;
                outputcontrol(delta);
             END;
          END;
pgdwn   : BEGIN    {Decrease by big amount}
             WITH modelparameters DO
             BEGIN
                CASE delta OF
                   1:BEGIN
                        Pfr := Pfr - 5*Pchange;
                        IF Pfr < 0 THEN Pfr := 0;
                     END;
```

```
                    2:BEGIN
                         Qfr := Qfr - 5*Qchange;
                         IF Qfr < 0 THEN Qfr := 0;
                      END;
                    3:BEGIN
                         BR := BR - 10*Brchange;
                         IF BR < 0.1 THEN BR := 0.1;
                      END;
                    4:BEGIN
                         Vtidal := Vtidal - 5*vtidalchange;
                         IF Vtidal < 0.025 THEN Vtidal := 0.05;
                      END;
                 END;
                 outputcontrol(delta);
              END;
           END;
uparrow : BEGIN      {Increase by small amount}
              WITH modelparameters DO
              BEGIN
                 CASE delta OF
                    1 :Pfr := Pfr + Pchange;
                    2 :Qfr := Qfr + qchange;
                    3 :BR := BR + Brchange;
                    4 :Vtidal := Vtidal + vtidalchange;
                 END;
                 outputcontrol(delta);
              END;
           END;
dnarrow : BEGIN      {Decrease by small amount}
              WITH modelparameters DO
              BEGIN
                 CASE delta OF
                    1:BEGIN
                         Pfr := Pfr - Pchange;
                         IF Pfr < 0 THEN Pfr := 0;
                      END;
                    2:BEGIN
                         Qfr := Qfr - Qchange;
                         IF Qfr < 0 THEN Qfr := 0;
                      END;
                    3:BEGIN
                         BR := BR - Brchange;
                         IF BR < 0.1 THEN BR := 0.1;
                      END;
                    4:BEGIN
                         Vtidal := Vtidal - vtidalchange;
                         IF Vtidal < 0.05 THEN Vtidal := 0.05;
                      END;
                 END;
                 outputcontrol(delta);
              END;
           END;
ltarrow : BEGIN     {Select new control variable to left}
              delta := delta - 1;
              IF delta < 1 THEN delta := 4;
              indicatecontrol(delta);
           END;
rtarrow : BEGIN     {Select new control variable to right}
              delta := delta + 1;
              IF delta > 4 THEN delta := 1;
              indicatecontrol(delta);
           END;
```

```
         END; {Case}
       END
      ELSE
       CASE ch OF
          #27 : BEGIN                    {Escape = quit}
                   gotoxy(1,25);
                   WRITELN(exitstring);
                   ch := readkey;
                   IF upcase(ch) = 'Y' THEN dosflag := TRUE;
                END;
       END;

       WHILE keypressed DO ch := readkey; {Clear buffer}

   UNTIL dosflag;
   leavegraphic;
END;

{-------------------------------------------------------}

PROCEDURE initialise_simulator;

{
Initialise the simulator default values and default parameters
to be plotted and dumped. Read default file if it exists and use
last defaults instead.
}

VAR i : INTEGER;
    b : barrecord;
    ch : CHAR;

BEGIN
   clrscr;
   inithelp('uctas.hlp');      {Initialise the help routine by reading}
                               {help info from UCTAS.HLP}
   dumpflag    := FALSE;
   xmax := 180;
   ymax := 2;
   speed := 60;
   glbgraph := false;   {Graph mode dormant}

   lotusarray[1] := p_Qfr;
   lotusarray[2] := p_Qalv;
   lotusarray[3] := p_Qe;
   lotusarray[4] := p_Pfr;
   lotusarray[5] := p_Pbc;
   lotusarray[6] := p_Pins;
   lotusarray[7] := p_Pl;      {Default dump file parameters}
   lotusarray[8] := p_Pa;
   lotusarray[9] := p_Q;
   lotusarray[10] := p_HR;
   lotusarray[11] := p_MABP;
   lotusarray[12] := p_BR;
   lotusarray[13] := p_Vtidal;

   midarray[1] := p_HR;
   midarray[2] := p_Q;
   midarray[3] := p_Sv;
   midarray[4] := p_MABP;
   midarray[5] := p_SystP;   {Middle screen parameters}
   midarray[6] := p_DiasP;
```

```
   IF memavail < 96000 THEN    {Check if enough memory}
   BEGIN
      clrscr;
      message(10,'You need at least 512K of memory to run this program');
      message(12,'Terminate all memory resident programs (eg sidekick)');
      message(14,'and try again.');
      message(20,'Terminate uctas now ? (Y/N) ..>');
      ch := readkey;
      IF upcase(ch) = 'Y' THEN halt;
   END;
   IF NOT(hardwarepresent) THEN {Check if a hercules card is present}
   BEGIN
      clrscr;
      message(10,'This program requires an IBM monochrome graphics card to run');
      message(12,'The graphics card was not detected.');
      message(20,'Terminate uctas now ? (Y?N) ..>');
      ch := readkey;
      IF upcase(ch) = 'Y' THEN halt;
   END;

   simulator;   {Run the simulator}

   WRITELN(dumpfil,'"  "');
   close(dumpfil);           {close the dumpfile before quitting}
   clrscr;
   message(10,'SAVE GRAPH AND LOTUS CHOICES AS DEFAULTS ? (Y/N)..>');
   glbch := READKEY;
   IF upcase(glbch) = 'Y' THEN
   BEGIN
      ASSIGN(defaultfil,'uctas.def');
      REWRITE(defaultfil);
      WITH defaultrecord DO
      BEGIN
         l := lotusarray;
         g := grarray;
         m := midarray;
         xm := xmax;          {Save defaults in UCTAS.DEF}
         ym := ymax;
         grfl := glbgraph;
         dpfl := dumpflag;
         sp := speed;
      END;
      WRITE(defaultfil,defaultrecord);
      CLOSE(defaultfil);
   END;
   clrscr;
END.
```