NIST

UNITED STATES
DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY

NIST Technical Note 1340

Strategy for Chemical Analysis of Alternative Refrigerants

Thomas J. Bruno

Strategy for Chemical Analysis of Alternative Refrigerants

Thomas J. Bruno

Thermophysics Division
Center for Chemical Technology
National Measurement Laboratory
National Institute of Standards and Technology
Boulder, Colorado 80303-3328

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 1990

CONTENTS

List of Tables

List of Figures

Strategy for Chemical Analysis of Alternative Refrigerants

Thomas J. Bruno

Thermophysics Division
National Institute of Standards and Technology
Boulder, CO 80303

The chemical purity of samples of alternative refrigerants which are used in the determination of thermophysical properties is of paramount importance. Many properties, such as vapor-liquid equilibria and transport properties such as thermal conductivity, require that measurements be performed on as pure a material as possible. Some common impurities such as water can have an extremely deleterious effect on a measurement, while the effects of other impurities are more subtle. Alternative refrigerants which are not available at purity levels that are normally considered "research grade" must be characterized in order for the data to be properly interpreted, and for the reported results to be used appropriately. In addition, it is necessary to assess the reactivity of alternative refrigerants in the presence of many of the common construction materials used in experimental apparatus. Many of the new fluids being considered as replacement refrigerants are known to be very aggressive towards metals and elastomers, especially at elevated temperatures and pressures. The potential for reaction must be considered before measurements are made in an expensive apparatus. During the course of studies on the thermophysical properties of alternative refrigerants, the National Institute of Standards and Technology has evolved an analytical protocol which is applied to each system. In this paper, the various analytical methods will be discussed, and the data of general usefulness presented. In addition, some of the novel instrumental approaches employed at NIST will be described.

Key words: alternative refrigerants; chemical analysis; gas chromatography; infrared spectrophotometry; mass spectrometry; water analysis.

1. INTRODUCTION

The threat of ozone depletion has led to a great deal of research to find suitable substitutes for some of the fully halogenated fluids which have been used for many years as refrigerants and propellants. Since some of these fluids will be phased out in many industrialized nations in the near

future, there is a pressing need for thermophysical property information on the more promising substitutes, which may be pure fluids or mixtures (binary and ternary) of fluids. The thermophysical properties that will be needed for effective design calculations include both equilibrium and transport properties. Some of the relevant equilibrium properties include the P-V-T surface, sound speed, critical parameters, refractive index, and surface tension. The important transport properties include thermal conductivity and viscosity.

It is of paramount importance that the experimental measurements of these properties be performed on well-characterized materials. This means that the purity of each fluid studied must be an important consideration at an early stage of the experimental design. It can often be difficult to obtain some fluids at the purity levels that are normally associated with "research grade" reagents. Further purification can be quite difficult and often requires large and costly distillation columns and zone refiners. The experimental results of most measurements cannot be properly interpreted without a detailed knowledge of the sample constituents. Under some circumstances, for example, a nominally single component fluid may have to be treated as a mixture in the analysis of results if the level of impurities is high.

Specific impurities which may be present in a sample may even make some measurements impossible. Water and air are two examples of contaminants that can plague an experimental measurement. Low levels of water can cause plugging of small-diameter transfer lines at lower temperatures. In addition, some measurements are seriously disturbed by the comparatively high electrical conductivity of water. An example is viscosity measurements based upon a torsional vibration quartz crystal. Air dissolved in fluids may cause vapor locking that can prevent the filling of measurement cells in many experimental apparatus. The presence of oils and lubricants in valves can also present a problem, since many of the alternative refrigerant fluids are moderately good

solvents, especially in the supercritical fluid state. These heavy hydrocarbon impurities can be present in the cylinder or container obtained from the supplier, or introduced inadvertently in the laboratory. Dissolved heavy hydrocarbons can be an especially vexing problem, since many oils will be moderately soluble at higher pressures (and densities), but will precipitate out of solution at atmospheric pressure. This can make detection of a heavy hydrocarbon contaminant very difficult, since the contaminant is usually left behind in a transfer line.

Another aspect of the problem is the effect of chemical decomposition and the reaction of the fluids with the materials of construction of measurement apparatus. It is very common for chemical degradation of the fluid to occur during the course of an experiment, especially under conditions of high temperature and pressure. Many of the alternative refrigerant fluids are very aggressive toward both metals and elastomers, especially at elevated temperatures. The thermal decomposition of these fluids can result in the formation of highly corrosive chemicals such as HF and HCl, and highly toxic materials such as carbonyl halides and carbon monoxide. Prudence dictates that the potential for inadvertently synthesizing such products be addressed under controlled conditions well in advance of the actual thermophysical property measurement. The possible catalytic effects of the construction materials of experimental apparatus should also be studied.

In this report, the general analytical protocol applied to alternative refrigerant fluids that are studied at the Thermophysics Division of the National Institute of Standards and Technology will be treated. The discussion will be divided into three parts; qualitative identification, quantitative determinations (especially those involving separation methods), and reaction screening. The qualitative identification discussion will focus on the verification of sample identity and the characterization of the major impurities that may be present. This process includes the identification of both positional and geometric isomers. Quantitation involves the optimization of analyses to provide the required level of accuracy in the determination of the concentrations of impurities [1]. This discussion will include some of the novel instrumental modifications developed for these

Obtain Required
Quantity of Fluid

Search Literature for Toxicology
and Safety Information:

Information Available?

yes

no

Provide Necessary
Precautions

Assume Fluid to be
Very Hazardous

Pre-measurement Analyses

Is fluid
gaseous?

no

yes

| Reaction Screening Tests | Verify Fluid Identity | Characterize Organic Impurities | Determine Water Content | Adsorbent Tests | Determine Air Content |

Figure 1. Analytical protocol for measurement work on alternative refrigerants.

analyses. The discussion of reaction screening will center around a protocol and apparatus developed in recent years for the support of high-temperature high-pressure thermophysical property measurements.

The analytical protocol in place at NIST is shown in flowchart form on figure 1. There are three major steps in the process; analysis of the starting materials, the measurement of the required thermophysical properties, and the analysis and characterization of the final contents of measurement apparatus. The strategy begins with the arrival (or synthesis) of the fluid, and always starts with a study of the hazards of the material. The material safety data sheet, several on-line databases, and published sources of information are consulted. The major sources are summarized in appendix 1. This study helps to provide an environment in which the materials can be handled safely, and is one of the most important steps in the process. If very little is known about a given fluid, a high level of hazard is assumed and the material is handled accordingly (glove box operations, protective clothing, restricted laboratory access, for example).

The next step is a battery of analytical tests applied to the samples before any thermophysical property measurements are performed. The identity of the sample is verified, and the major impurities (including organics and air) are characterized. Since most refrigerant fluids are hygroscopic, water analysis, either using a chromatographic method or the Karl Fischer titration, is always performed. Adsorbent tests are done to assess the presence of heavy contaminants. When measurements under severe conditions of temperature and pressure are anticipated, reaction screening is part of the initial analytical step. The reaction screening process includes the study of possible catalytic effects due to the construction materials of the measurement apparatus.

After the property measurements have been made, the sample is recovered (if possible) and an additional series of tests is performed. Sample recovery is usually done by condensing the fluid in stainless steel bulbs cooled in liquid nitrogen. The possibility of chemical decomposition during a measurement requires that each sample of fluid be recovered and checked for organics. To assess

the possibility of a fluid being inadvertently contaminated in an apparatus, water analysis and adsorbent tests are also redone. In addition, it is sometimes necessary to recheck for air, since a leak in an apparatus may not be detected in any other way. A summary of results from the analyses are always included in the report of any measured properties.

2. QUALITATIVE IDENTIFICATION

The two most useful instrumental techniques for the qualitative identification of alternative refrigerant components are mass spectrometry and infrared spectrophotometry. When possible, it is highly desirable to have both measurements available for study, although mass spectrometry is by far the more important of the two. A study of the spectra of about sixty chlorofluoro, chloro-, and chlorofluorobromoethanes and -ethylenes has led to the development of a spectral data library [2], as well as the recognition of some of the general spectral characteristics of these molecules. A listing of the materials studied is provided in table 1. In each case, the highest purity material commercially available was obtained for study.

Table 1. List of compounds studied in the development of spectral library and in the study of spectral trends

	Codes	Compounds
Ethane Compounds	112	1,2-difluorotetrachloroethane
	112a	1,1-difluorotetrachloroethane
	113	1,1,2-trichlorotrifluoroethane
	113a	1,1,1-trichloro-2,2,2-trifluoroethane
	114	1,2-dichlorotetrafluoroethane
	113a	1,1-dichlorotetrafluoroethane
	115	chloropentafluoroethane
	116	hexafluoroethane
	121	1,1,2,2-tetrachlorofluoroethane

Table 1. List of compounds studied in the development of spectral library and in the study of spectral trends (contd.)

Codes	Compounds
122	1,1-difluoro-1,2,2-trichloroethane
123	2,2-dichloro-1,1,1-trifluoroethane
123a	1,2-dichlorotrifluoroethane
124	2-chloro-1,1,1,2-tetrafluoroethane
125	pentafluoroethane
131	1,1,2-trichloro-2-fluoroethane
131a	1,1,2-trichloro-1-fluoroethane
132b	1,2-dichloro-1,1-difluoroethane
133a	2,2,2-trifluoroethyl chloride
134	1,1,2,2-tetrafluoroethane
134a	1,1,1,2-tetrafluoroethane
141	1,2-dichlorofluoroethane
141b	1,1-dichloro-1-fluoroethane
142b	1-chloro-1,1-difluoroethane
143	1,1,2-trifluoroethane
143a	1,1,1-trifluoroethane
152a	1,1-difluoroethane
161	fluoroethane
113B2$\alpha\beta$	2-chloro-1,2-dibromo-1,1,2-trifluoroethane
113B2	1-chloro-1,1-dibromotrifluoroethane
114B2	1,2-dibromotetrafluoroethane
123B1	2-bromo-2-chloro-1,1,1-trifluoroethane
123B2	1,2-dibromo-1,1,2-trifluoroethane
123aB1α	1-bromo-2-chloro-1,1,2-trifluoroethane
132bB2	1,2-dibromo-1,1-difluoroethane
133aB1	2,2,2-trifluoroethyl bromide
142B1	2-bromo-1,1-difluoroethane

8

Table 1. List of compounds studied in the development of spectral library and in the study of spectral trends (contd.)

	Codes	Compounds
	151B1	1-bromo-2-fluoroethane
Ethylene Compounds	1110	tetrachloroethylene
	1111	trichlorofluoroethylene
	1112a	1,2-dichloro-2,2-difluoroethylene
	1113	chlorotrifluoroethylene
	1114	tetrafluoroethylene
	1120	trichloroethylene
	1121	1,2-dichloro-1-fluoroethylene
	1123	trifluoroethylene
	1130	1,2-dichloroethylene (cis- and trans-)
	1130a	1,1-dichloroethylene
	1131a	1-chloro-1-fluoroethylene
	1132a	1,1-difluoroethylene
	1141	fluoroethylene
	1112aB2	1,1-dibromodifluoroethylene
	1122B1	1-bromo-2,2-difluoroethylene

2.1 Mass Spectra

The interpretation of a mass spectrum usually begins with the parent or molecular ion, if it is present in the spectrum. The parent ion is the fragment resulting from the loss of an electron from the molecule, and provides the relative molecular mass of the compound. The isotopic cluster of the parent ion chlorofluorobromoethane and -ethylenes is rich with information, due to the natural abundances of the relevant isotopes. Table 2 shows the more important isotopes, along with the percent abundance [3]. Chlorine and bromine are commonly called M + 2 elements (where M is the parent or molecular ion), because a relatively abundant heavier isotope is naturally present. This

Table 2. Natural abundance of relevant isotopes[a]

Element	Total no. of isotopes	More prominent isotopes (mass percent abundance)
Hydrogen	3	$^{1}H(1.00783, 99.985)$ $^{2}H(2.01410, 0.015)$
Carbon	7	$^{12}C(12.00000, 98.9)$ $^{13}C(13.00335, 1.1)$
Fluorine	6	$^{19}F(18.99840, 100.0)$
Chlorine	11	$^{35}Cl(34.96885, 75.5)$ $^{37}Cl(36.96590, 24.5)$
Bromine	17	$^{79}Br(78.9183, 50.5)$ $^{81}Br(80.91642, 49.5)$

[a] Taken from T.J.Bruno, P.D.N. Svoronos, Handbook of Basic Tables for Chemical Analysis, CRC Press, Boca Raton, 1989.

results in a peak multiplicity that is a significant help in spectrum interpretation. Table 3 presents calculated values of the abundances of the parent ion isotopic cluster for various combinations of chlorine and bromine [3]. The accuracy of the predictions is typically between 5 and 10 percent; the agreement of experimental data with the values in the table greatly depends upon the instrument used to record the spectra.

It is often convenient to measure the mass spectrum of some alternative refrigerant fluids in a suitable solvent. Samples which are gaseous at room temperature can often be dissolved in carbon tetrachloride, toluene, or some other solvent, especially at temperatures near 0 °C. In addition, some chlorofluorobromoethanes and -ethylenes are very sensitive to both air and moisture. Dissolving the sample in an anhydrous solvent provides a margin of safety in the study of some of these compounds, and allows the use of gas chromatography-mass spectrometry procedures. The important peaks in

Table 3. Relative intensities of isotope peaks for combinations of bromine and chlorine[a] (P designates the parent or molecular ion.)

		Br_0	Br_1	Br_2	Br_3	Br_4
Cl_0	P + 2		98.0	196.0	294.0	390.8
	P + 4			96.1	288.2	547.7
	P + 6				94.1	375.3
	P + 8					92.0
Cl_1	P + 2	32.5	130.6	228.0	326.1	424.6
	P + 4		31.9	159.0	383.1	704.2
	P + 6			31.2	187.4	564.1
	P + 8				30.7	214.8
	P + 10					30.3
Cl_2	P + 2	65.0	163.0	261.1	359.3	456.3
	P + 4	10.6	74.4	234.2	490.2	840.3
	P + 6		10.4	83.3	312.8	791.6
	P + 8			10.2	91.7	397.5
	P + 10				9.8	99.2
	P + 12					10.1
Cl_3	P + 2	97.5	195.3	294.0	393.3	
	P + 4	31.7	127.0	99.7	609.8	
	P + 6	3.4	34.4	159.4	473.8	
	P + 8		3.3	37.1	193.9	
	P + 10			3.2	39.6	
	P + 12				3.0	
Cl_4	P + 2	130.0	228.3	326.6	4.2	
	P + 4	63.3	190.9	414.9	735.3	
	P + 6	13.7	75.8	263.1	670.0	
	P + 8	1.2	14.4	88.8	347.1	
	P + 10		1.1	15.4	102.2	
	P + 12			1.3	16.2	
	P + 14				0.7	
Cl_5	P + 2	162.6	260.7	358.9		
	P + 4	105.7	265.3	520.8		
	P + 6	34.3	137.9	397.9		
	P + 8	5.5	39.3	174.5		
	P + 10	0.3	5.8	44.3		
	P + 12		0.3	5.7		
	P + 14			0.5		
Cl_6	P + 2	195.3				
	P + 4	158.6				
	P + 6	68.8				
	P + 8	16.6				
	P + 10	2.1				
	P + 12	0.1				

[a] Taken from T.J. Bruno, P.D.N. Svoronos, Handbook of Basic Tables for Chemical Analysis, CRC Press, Boca Raton, 1989.

Table 4. Important peaks in the mass spectra of common solvents[a]

Solvents	Formula	P$^+$	Important peaks (m/e)
Water	H_2O	4̸8(100%)	17
Methanol	CH_3OH	32	31(100%),29,15
Acetonitrile	CH_3CN	41(100%)	40,39,38,28,15
Ethanol	CH_3CH_2OH	46	45,31(100%),27,15
Dimethyl ether	CH_3OCH_3	46(100%)	45,29,15
Acetone	CH_3COCH_3	58	43(100%),42,39,27,15
Acetic acid	CH_3CO_2H	60	45,43,18,15
Ethylene glycol	$HOCH_2CH_2OH$	62	43,33,31(100%),29,18,15
Furan	C_4H_4O	4̸8(100%)	42,39,38,37,29,18
Tetrahydrofuran	C_4H_8O	72	71,43,42(100%),41,40, 39,27,18,157,18,15
n-Pentane	C_5H_{12}	72	57,43(100%),42,41,39, 29,28,27,15
Dimethylformamide (DMF)	$HCON(CH_3)_2$	73(100%)	58,44,42,30,29,28,18,15
Diethylether	$(C_2H_5)O_2$	74	59,45,41,31(100%),29, 27,15
Methyl acetate	$CH_3CO_2CH_3$	74	59,43(100%),42,32,29, 28,15
Carbon disulfide	CS_2	76(100%)	64,44,38,32
Benzene	C_6H_6	76(100%)	77,52,51,50,39,28
Pyridine	C_5H_5N	79(100%)	80,78,53,52,51,50,39,26
Dichloromethane	CH_2Cl_2	84	86,51,49(100%),48,47, 35,28
Cyclohexane	C_6H_{12}	84	69,56,55,43,42,41,39,27
n-Hexane	C_6H_{14}	86	85,71,69,57(100%),43, 42,41,39,29,28,27
p-Dioxane	$C_4H_8O_2$	88(100%)	87,58,57,45,43,31,30, 29,28
Tetramethylsilane (TMS)	$(CH_3)_4Si$	88	74,73,55,45,43,29
1,2-Dimethoxy ethane	$(CH_3OCH_2)_2$	90	60,58,45(100%),31,29

Table 4.

Table 4. Important Peaks in the mass spectra of common solvents[a] (contd.)

Solvents	Formula	P[+]	Important peaks (m/e)
Toluene	$C_6H_5CH_3$	92	91(100%),65,51,39,28
Chloroform	$CHCl_3$	118	120,83,81(100%),47,35, 28
Chloroform-d$_1$	$CDCl_3$	119	121,84,82(100%),48,47, 35,28
Carbon tetrachloride	CCl_4	152 (not seen)	121,119,117(100%),84, 82,58.5,47,35,28
Tetrachloroethene	$CCl_2=CCl_2$	164 (not seen)	168,166(100%),165,164, 131,128,129,95,94,82, 69,59,47,31,24

P[+] = parent ion.

[a] Taken from T.J. Bruno, P.D.N. Svoronos, <u>Handbook of Basic Tables for Chemical Analysis</u>, CRC Press, Boca Raton, 1989.

the mass spectra of most of the commonly used solvents are summarized in table 4 [3]. This table is used to distinguish the spectrum of the refrigerant from that of the solvent. As usual, a solvent blank should be measured to guard against the misinterpretation of impurity peaks present in the spectrum of the solvent itself.

The fragmentation patterns of chlorofluorobromoethanes and -ethylenes are, in general, relatively simple. Complex rearrangement reactions are rarely seen, and the fragmentation is usually straightforward. The kind of fragmentation that will occur under electron impact conditions depends upon the relative bond strengths of the moieties that are summarized in table 5 [4]. These data can be used to judge the relative importance of possible fragmentation mechanisms. The bonds between carbon and the heavier halogens (chlorine and bromine) are relatively weak, and fragmentation will be favored at these sites. The carbon-fluorine bond is considerably stronger than the others, and

Table 5. Chemical bond energies of relevance to alternative refrigerants

Bond	Bond Energy kcal/g–bond
C – H	98.7
C – C	82.6
C = C	145.8
C ≡ C	199.6
C – F	116
C – Cℓ	81
C – Br	68

therefore fragments of these bonds will be much less intense, if they are observed at all. These fragments will produce especially weak peaks if the molecule has a fluorine(s) and a heavier halogen(s) present. The energy absorbed in the electron impact process will be dissipated in breaking the weaker bonds rather than the C-F bond. In most cases of this simple cleavage, the hydrocarbon fragment will generally take the positive charge and the halogen fragment will become the neutral radical species. The only common exception to this halocarbon fragmentation mechanism occurs with iodine containing compounds. In these cases, the iodine usually retains the charge. The simple rearrangements producing HCℓ and HF fragments can also occur, but the formation of HBr is not as likely as the loss of Br from the molecule. The occurrence of the reaction producing the HF species is more likely than those producing either HCℓ or HBr, due to the more favorable negative heat of formation of HF.

By measuring the mass spectrum of each of the materials listed in table 1, a compilation of the common fragments lost from typical refrigerants has been developed [2]. The ion abundances

have been normalized, and the expected peak intensity for each fragment has been calculated. These fragments, which are summarized in table 6, are very useful in the analysis of an unknown spectrum. The mass/charge values provided are for low resolution conditions. Users having access to high resolution mass spectrometers can make adjustments in these values with the aid of table 2. Fragments containing one or more bromine atoms are tabulated with the slightly more abundant ^{79}Br isotope only. The reader should expect to find the corresponding signal due to ^{81}Br in addition to that from the ^{79}Br. The expected intensity of each fragment peak is rated from "strong" to "weak." When a range of intensity is expected, an asterisk often denotes the most commonly observed intensity. The actual instrumental conditions used to measure an individual mass spectrum have a significant influence on the absolute ion abundances that will be measured. For this reason, a more quantitative prediction is not warranted. In addition to instrumental factors, one must also consider the influence of the "departing species" that leads to the formation of a fragment. As an example, we can consider the fragment at mass/charge = 63, identified as $CH_2CH_2C\ell$, having an intensity ranging from medium to strong. The intensity will be stronger if the fragment results from the loss of a bromine, but much less intense if a fluorine must be lost to form the fragment. The mass/charge values listed in the table are those determined from the most abundant isotopes of the atoms constituting the fragment. The other relevant isotopic peaks will also be present in the mass spectrum and are an aid in identification using tables 2 and 3 [4-6].

The user should always look for the presence of confirmatory peaks, especially when two or more fragments are known to give peaks at a given mass/charge value. As an example, at a mass/charge value of 117, two possible fragments are 1) $CF_2CFC\ell H$ and 2) $CF_3CHC\ell$. If, in addition, a strong peak is found at mass/charge = 69 (indicative of the CF_3 fragment), one can be

Table 6. Common refrigerant molecule fragments

Mass/Charge	Fragment	Intensity
19	F	medium
27	HC=CH$_2$	medium
33	CH$_2$F	medium
35	Cℓ	strong
36	HCℓ	medium*-strong
45	FC=CH$_2$ HC=CHF	strong strong
47	FCH$_2$CH$_2$ CH$_3$CHF	strong strong
48	CHCℓ	weak
49	CH$_2$Cℓ	medium
50	CF$_2$	weak*-medium
51	CHF$_2$	strong
60	HC≡CCℓ	medium
61	HC=CHCℓ CℓC=CH$_2$	strong strong
63	CH$_2$CH$_2$Cℓ F$_2$C=CH CH$_3$CHCℓ HFC=CF	medium-strong medium medium medium
64	FHC=CHF F$_2$C=CH$_2$	weak*-medium weak*-medium
65	CH$_2$CHF$_2$ CH$_3$CF$_2$ CH$_2$FCHF	medium medium-strong medium
66	FCCℓ	medium
67	CFCℓH	weak-strong
69	CF$_3$	medium-strong*
78	CℓC≡CF	weak*-medium
79	Br FC=CHCℓ HC=CFCℓ CℓC=CHF	medium-strong medium medium medium

16

Table 6. Common refrigerant molecule fragments (contd.)

Mass/Charge	Fragment	Intensity
80	HBr	weak*-medium
81	$CH_3CC\ell F$ $CH_2C\ell CFH$ $C\ell CHCH_2F$ $F_2C=CF$ ^{81}Br	strong medium-strong medium-strong medium medium
82	$F_2C=CHF$ $CC\ell_2$	weak-medium weak*-medium
83	CH_2CF_3 CHF_2CHF $CHC\ell_2$ CF_2CH_2F	strong strong strong strong
85	$CC\ell F_2$	strong
93	CH_2Br	weak
94	$C\ell C\equiv CC\ell$	weak*-medium
95	$CH=CC\ell_2$ $C\ell C=CHC\ell$	medium*-strong medium*-strong
97	$CH_2C\ell CHC\ell$ $C\ell FC=CF$ $CHC\ell_2CH_2$ $CH_3CC\ell_2$	weak weak-medium weak-medium weak-medium
98	$F_2C=CHC\ell$ $FHC=CFC\ell$	weak-medium weak-medium
99	$CH_2C\ell CF_2$ $CF_2HCHC\ell$ $CH_2FCFC\ell$ $CHFC\ell CHF$ $CF_2C\ell CH_2$	weak-strong weak-strong weak-strong weak-strong weak-strong
100	$F_2C=CF_2$	weak-medium
101	$CC\ell_2F$ CF_2HCF_2 CF_3CHF	strong weak weak-medium
113	$C\ell C=CC\ell F$ $C\ell_2C=CF$	weak*-medium weak

Table 6. Common refrigerant molecule fragments (contd.)

Mass/Charge	Fragment	Intensity
115	$CH_2ClCFCl$	weak
	$CHCl_2CFH$	weak
	CH_2FCCl	weak
	$CClFCH_2$	weak
	$CHFClCHCl$	weak
117	CCl_3	medium-strong
	CF_2ClCHF	medium
	CF_2CFClH	medium
	CF_3CHCl	strong
	CF_2HCFCl	medium
119	CF_3CF_2	medium
123	$HBrC=CF$	weak
	$FBrC=CH$	weak
	$HFC=CBr$	weak
125	$CHFCH_2Br$	weak
	CH_2CHFBr	weak
	$CHBrCH_2F$	weak
	$CBrFCH_3$	weak
129	CF_2Br	weak-medium
	$Cl_2C=CCl$	medium
131	HCl_2CHCl	weak
	CCl_3CH_2	weak
	$CH_2ClCHCl$	weak
133	CF_2CHCl_2	medium
	$CF_2ClCClH$	medium
	CCl_2CHF_2	medium
	$CHFCFCl_2$	medium
135	$CClFCF_3$	strong
	CF_2CF_2Cl	medium
141	$BrC=CF_2$	weak
	$FC=CFBr$	weak
143	CF_2CH_2Br	weak-medium
	$CFBrCH_2F$	weak-medium
145	$CClFBr$	medium
151	CCl_2FCF_2	medium
	$CFClCF_2Cl$	medium
	CCl_2CF_3	medium

Table 6. Common refrigerant molecule fragments (contd.)

Mass/Charge	Fragment	Intensity
161	CF_2CFBrH $CHFCBrF_2$ $CHBrCHF_2$	strong strong strong
167	$CF_2C\ell CC\ell_2$ $CC\ell_3CF_2$	weak-medium weak-medium
177	$CF_2CBrC\ell H$	weak-medium
179	BrF_2CCF_2 CF_3CFBr	strong strong
195	$CBrC\ell CF_3$ $CF_2CFBrC\ell$ $CFC\ell CF_2Br$ $CFBrCF_2C\ell$	strong strong medium medium
205	$CC\ell Br_2$	weak

* Indicates the more commonly observed intensity.

more confident that the peak at 117 is due to fragment 2. One should also look for the appearance of the "departing species" elsewhere in the mass spectrum when deciding on the assignment of a particular peak.

A computer program that assists the user in the interpretation of the mass spectrum of an unknown has been developed, and is designed to be especially useful for chlorofluorobromoethanes and -ethylenes. A listing of this program is provided in appendix 2. The program provides possible combinations of atoms which can form a given mass/charge value for a fragment. A search of common refrigerant fragments, and an analysis of the molecular ion cluster is also included. The program is written in Turbo Pascal (5.0)[1], and requires at least 256k of memory to run on a personal computer.

---

[1]Certain commercial equipment, instruments, or materials are identified in this paper only in order to specify adequately the experimental procedure. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

2.2 Infrared Spectrophotometry

Infrared spectrophotometry can often be very useful in the qualitative identification of refrigerants and refrigerant impurities [7,8]. It is also possible, of course, to use the Beer-Lambert law with infrared absorption or transmission measurements to obtain quantitative data having an accuracy of between 3 and 5 percent [3]. In this section, the discussion will focus on several useful spectral regions that provide information on the structure (and therefore the identity) of the refrigerant.

The first region to note in the infrared region is the C-H stretching (symmetric and asymmetric) band that occurs between 2850 and 3100 cm^{-1}. If absorptions occur in this region, they are usually very intense due to the large change in dipole moment that results during the C-H vibration. The presence of these bands in a spectrum is, of course, clear evidence of a refrigerant which is not fully halogenated. Conversely, the absence of bands centered around 3000 cm^{-1} is evidence of a fully halogenated refrigerant. If the carbon of the C-H band also carries a chlorine atom, the absorption is often shifted to a higher energy, usually centered at 3050 cm^{-1} sometimes extending as far as 3100 cm^{-1}. This effect sometimes occurs when the carbon carries a fluorine or bromine atom as well, but the assignment of the band in these cases is less reliable. In this respect, the C-H region of the infrared spectra of refrigerants can sometimes resemble those of aromatic hydrocarbons. When a C-H band is observed, the multiplicity of the envelope is also worthy of note. When the band is due to a carbon which carries a single hydrogen, the absorption will be a single peak which is usually sharp. When the band is from a carbon having two hydrogens, it almost always occurs as a sharp doublet. When, in addition, the carbon also carries a chlorine or bromine, one of the peaks of the doublet will usually be shifted to a higher energy (above 3000 cm^{-1}).

The double bond region of the infrared spectrum occurs between 2000 and 1500 cm^{-1}. In this range, a medium to strong absorption centered at 1650 cm^{-1} is evidence of an ethylene rather than an ethane based compound. The intensity and shape of this band provides an indication of the degree of symmetry of the molecule. Since the change in dipole moment of the $>C=C<$ stretching motion depends largely on the substituents, a relatively strong, broad absorption indicates a highly asymmetric molecule. On the other hand, a weaker, thinner peak indicates a more nearly symmetric molecule. Naturally, a molecule with mirror plane symmetry with respect to the double bond will have no change in dipole moment due to the $>C=C<$ vibration. These vibrations will be inactive in the infrared (the transitions being forbidden by quantum mechanical considerations), and will produce no absorptions or absorptions which are very weak.

The characteristic absorptions from carbon-halogen bonds are summarized in table 7 [3]. These absorptions occur in what is commonly referred to as the fingerprint region, since many absorptions which are found between 1500 and 600 cm^{-1} are characteristic of strongly correlated whole-molecule vibrations. Since most single bonds absorb at similar frequencies, the vibrations tend to couple. These vibrations complicate this region, and make the carbon-halogen peaks difficult to interpret. Furthermore, the absorptions due to C-Br stretching often lie outside the range of many infrared spectrophotometers. The C-F vibrations produce a useful overtone envelope between 1900 and 2600 cm^{-1}. The number of peaks and the complexity of the envelope increase as the number of fluorine atoms on the molecule increases.

The region between 3700 and 3200 cm^{-1} contains absorptions due to water contamination. Very high levels of water will produce the familiar broad band of medium intensity, while lower levels (down to the detectable limit of about 0.1 percent) produce a thinner, less intense peak shifted to slightly higher energy.

Table 7. Infrared absorptions of carbon-halogen bonds[a]

General formula	>C-X stretch	>CX$_2$ stretch	-CX$_3$ stretch	C-X stretch
X = F	1120-1010	1350-1200 (asym) 1200-1080 (sym)	1350-1200 (asym) 1200-1080 (sym)	1230-1100
X = Cℓ	830-500 1510-1480 (overtone)	845-795 (asym) ~ 620 (sym)		
X = Br	667-290			

[a] Taken from T.J. Bruno, P.D.N. Svoronos, <u>Handbook of Basic Tables for Chemical Analysis</u>, CRC Press, Boca Raton, 1989.

2.3   Adsorbent Tests

The problem of heavy hydrocarbons dissolved in alternative refrigerant fluids was mentioned in the introduction. Many of the fluids themselves are moderately strong solvents, capable of picking up surprising amounts of oils. It may be impossible to detect these contaminants by analyzing the sample in the vapor phase under ambient conditions, since the solubility of such hydrocarbons in the fluid will drop significantly at lower fluid densities. A simple way to address the possibility of heavy hydrocarbon contamination is to pass the fluid through a stainless steel cartridge packed with a suitable adsorbent (or sorbent, as they are also called), as shown in figure 2a. At first glance, it may seem unusual to use trapping adsorbents in the study of heavier contaminants, since they have mainly been used in the analysis of volatiles in liquids such as water. A list of the more common adsorbents is provided in table 8 [3]. In addition to these materials, many of the stationary phases used in gas-solid chromatography may be used in adsorption traps [3].

solvent
resevoir

Pyrex wool plugs

stainless steel
tube

Tenax - GC

to collection
vial
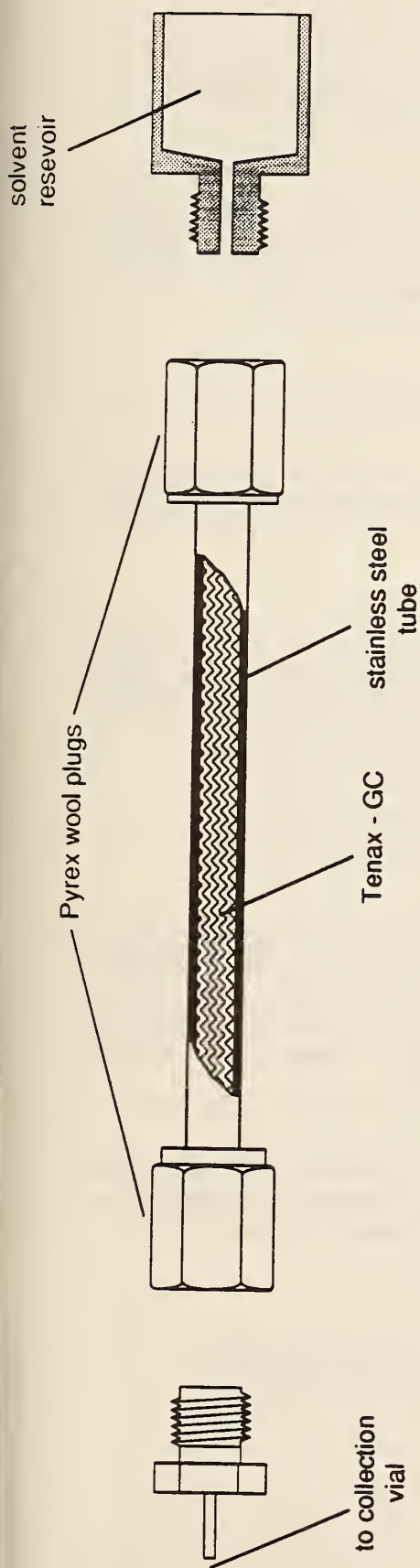
Adsorbent Cartridge

(a)
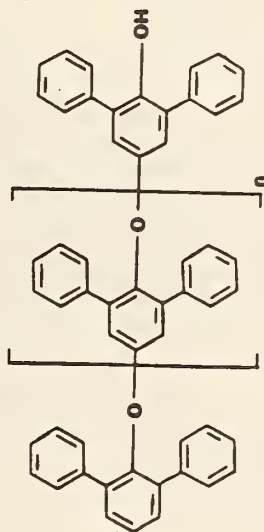
Tenax - GC

(b)

Figure 2a. The typical trapping adsorbent cartridge.
Figure 2b. The chemical structure of Tenax-GC.

23

Table 8. Trapping Adsorbents*

Adsorbent	Desorption solvents	Applications
Activated carbon	Carbon disulfide, methylene chloride, diethyl ether, diethyl ether with 1% methanol, diethyl ether with 5% 2-propanol (caution: CS_2 and CH_3OH can react in the presence of charcoal).	Used for common volatile organics; examples include methylene chloride, vinyl chloride, chlorinated aliphatics, aromatics, and acetates.

Note: Metallic or salt impurities in the adsorbent can sometimes cause the irreversible adsorption of electron-rich oxygen functionalities; examples include 1-butanol, 2-butanone, and 2-ethoxyacetate; recovery rate is often poor for polar compounds.

Graphitized carbon-black	Carbon disulfide, methylene chloride, diethyl ether.	Used for common volatile aliphatic and aromatic compounds, organic acids and alcohols, and chlorinated aliphatics.

Note: These adsorbents are hydrophobic and are not very sensitive to moisture.

Silica gel	Methanol, ethanol, water, diethyl ether.	Used for polar compound collection and concentration; examples include alcohols, phenols, chlorophenols, chlorinated aromatics, aliphatic and aromatic amines, and nitrogen dioxide.

Note: Useful for compounds which cannot be recovered from the charcoal adsorbents; the most serious problem with silica is the effect of water, which can cause desorption of the analytes of interest; the heating effect involved can sometimes initiate reactions such as polymerization of the analyte.

* Taken from T.J. Bruno, P.D.N. Svoronos, Handbook of Basic Tables for Chemical Analysis, CRC Press, Boca Raton, 1989.

Table 8. Trapping Adsorbents (contd.)

Adsorbent	Desorption solvents	Applications
Activated alumina	Water, diethyl ether, methanol.	Used for polar compounds such as alcohols, glycols, ketones, and aldehydes; has also been used for polychlorinated biophenyls and phthalates.

Note: Similar in application to silica gel.

Porous polymers	Hexane, diethyl ether, alcohols.	Used for a wide range of compounds which include phenols, acidic and basic organics, pesticides, and priority pollutants.

Note: The most commonly used porous polymer sorbent is Tenax-GC, although the Porapak and Chromosorb Century series have also been used; Tenax-GC has been used with thermal desorption methods, but can release toluene, benzene, and trichloroethylene residues at the higher temperatures; in addition to Tenax-GC, XAD-2-8, Porapak-N, and Chromosorbs 101, 102, 103, and 106 have found applications, sometimes in "stacked" sampling devices (for example, a sorbent column of Tenax-GC/Chromosorb 106 in tandem); Chromosorb 106, a very low-polarity polymer, has the lowest retention of water with respect to organic materials, and is well suited for used as a back-up sorbent.

Bonded phases	Methanol, hexane, diethyl ether.	Used for specialized applications in pesticides, herbicides, and polynuclear aromatic hydrocarbons.

Note: Most expensive of the common adsorbents; useful for the collection of organic samples from water.

Molecular sieves	Carbon disulfide, hexane, diethyl ether.	Have been used for the collection of aldehydes and alcohols, and for acrolein.

Note: Molecular sieve 13-X is the main molecular sieve to be used as a trapping adsorbent; these adsorbents will also retain water.

The most commonly used adsorbent is Tenax-GC. Tenax-GC is a linear polymer of p-2-6-diphenyl-phenylene oxide, the chemical structure of which is shown in figure 2b. Although the surface area of Tenax-GC is lower than most of the other porous polymer adsorbents (such as the Porapak and Century materials), the high thermal stability makes it easy to activate, especially for repetitive usage in the collection of heavier hydrocarbons. It must be noted that many of the other adsorbents in table 8 will adsorb the refrigerants themselves. For example, the common molecular sieve materials strongly and exothermically adsorb chlorofluorocarbons. This is usually not a problem with any of the porous polymer materials, however some of them are reactive toward nitrogen-containing functionalities such as amines. Tenax-GC is not attacked by any common chemical functionalities.

The conditions in the adsorbent trap should be chosen to mimic those in the sample container. This is easily accomplished in the case of fluids consisting of liquid under its own vapor pressure. The fluid is allowed to flow through the cartridge until a stream of liquid appears at the exit. At this point, the flow is decreased, so as to create a pressure gradient (and therefore a density gradient) through the length of the cartridge, resulting in fluid vaporization inside the adsorbent bed. In the case of more volatile or gaseous fluids, a capillary restrictor may be needed to maintain a density gradient through the cartridge. The heavier hydrocarbon contaminants will precipitate out on the adsorbent. The cartridge is then detached from the fluid container, and the adsorbent is extracted with a few milliliters of solvent (such as toluene or carbon tetrachloride) to desorb and dissolve any contaminants that may have been collected.

The extracted solution should be analyzed using gas chromatography-mass spectrometry. Pump oils, lubricants and machining fluids are all common contaminants that can be inadvertently introduced in various stages of the measurement process. The identification of these impurities can be much easier using a disk-based contaminant library consisting of the mass spectra of the major chromatographic peaks of common oils and liquids found in the laboratory.

3. QUANTITATION AND SEPARATION METHODS

Gas chromatography, the most generally useful of all separation methods available in chemical analysis, is also the workhorse in the analysis of alternative refrigerant fluids. Applications involving both capillary and packed column methods are of value. The response factors of many alternative refrigerants to the more useful chromatographic detectors are often very similar, especially for a group of closely related isomers. This fact can be of assistance in calibration procedures, especially when the highest quantitative accuracy is not required.

3.1 Analysis of Organics

The use of capillary columns is especially valuable in the analysis of the organic (or chloro-fluoro-bromo carbon) components of refrigerants. Much of the published work on trace analysis of refrigerants in the atmosphere describes the application of specialized packed columns. Packed columns are useful in trace analysis for a variety of reasons, among them the ability to apply large samples to the column without causing overload, and the delivery of the entire sample to the detector (ie., no sample splitting). The analysis of relatively pure fluids in the laboratory as part of an overall measurement program is the opposite end of the analytical spectrum. There is usually no need to quantitate at the part per trillion (by volume) level. Impurities present at the level of hundreds of parts per million are enough to seriously influence thermophysical property measurements.

Retention on capillary columns is a complex function of many factors such as column temperature, stationary phase film composition and thickness, and carrier gas properties, to name only a few. One of the most important physical parameters that the chemist can set is the stationary phase film thickness. Standard capillary columns have a thickness between 0.1 and 0.3 μm. Coatings that are considered thick range from 0.3 to approximately 1 μm. In recent years, very thick-film columns,

having coatings ranging between 1 and 8 μm have been introduced, and have met with success in certain difficult analyses. The theoretical predictions of a dramatic drop in column efficiency using very thick stationary phase films have not been borne out by experiment. Thick and very thick column coatings are of value in applications that require relatively high loading capacity, and where a high level of retention is needed to achieve separation. An important example is in the analysis of very volatile materials and gases.

Thick film and very thick film stationary phase coatings (having film thicknesses of between 1 and 3 μm) are usually needed to provide good resolution of the isomeric impurities in refrigerant fluids. The most useful stationary phase for capillary column gas chromatography has been the boiling point columns based on polymethylsiloxane. When very subtle differences exist among isomers, subambient temperature operation is often necessary. Although it is possible to pump a cryogen into the column oven, good results can be obtained using a Ranque-Hilsch vortex tube to introduce a stream of cold air [9]. Temperatures as low as -40 °C can be obtained using this device, and temperature equilibration can be accomplished in less than 30 minutes.

When the most sensitive detectors (such as mass spectrometers and electron capture detectors) are used with capillary columns, a split/splitless injection system is usually required. This type of injector is required in order to avoid overloading the column or detector. In normal practice, sample is applied to this injector using a chromatographic syringe. This presents minimal difficulty with samples of low volatility. More volatile samples often present serious difficulties with sample evaporation and sample discrimination, usually making the use of split/splitless injectors impossible. While under some limited circumstances it may be possible to adapt a valve/splitter arrangement (such as the injectors used in supercritical fluid chromatography), such an arrangement is nonstandard and expensive. In any case, a valve splitter combination can only be used when there is ample supply of sample.

The simple apparatus shown in figure 3 can often be used to eliminate the difficulty in handling moderately volatile samples [10]. The cold air stream from a vortex tube is directed into a glass jacket which surrounds a standard chromatographic syringe. It is then possible to chill the syringe and its contents to temperatures as low as -40 °C. This will significantly reduce the vapor pressure of many of the alternative refrigerant fluids and allow them to be injected in the liquid state. The problems associated with sample discrimination in analyses of less volatile refrigerants are also dramatically reduced. A more complete description of the chilled syringe and examples of its use can be found in appendix 3.

The most valuable high sensitivity detectors for use in capillary gas chromatography of alternative refrigerants are the electron capture detector and the mass spectrometer (or mass selective detector). Although it can often be difficult to optimize, the electron capture detector provides the best sensitivity to fluids containing many chlorine and bromine atoms. The mass selective detector has the advantage of providing structural information through the mass spectrum; however, for quantitative measurements, it is usually better to monitor the current of a single abundant ion. When used in the single ion monitoring mode, the mass selective detector can approach the sensitivity of the electron capture detector [1].

Quantitative analysis can also be applied to contaminants extracted from adsorbent columns, a technique that was discussed earlier. It is important to use solvent rather than thermal desorption methods, since the recovery rates are usually much higher when the contaminants are dissolved in an appropriate solvent. Although it is usual to use only the minimum quantity of solvent to extract the adsorbent, preconcentration of the solution in a rotary evaporator may be employed if adequate sensitivity is not obtained.
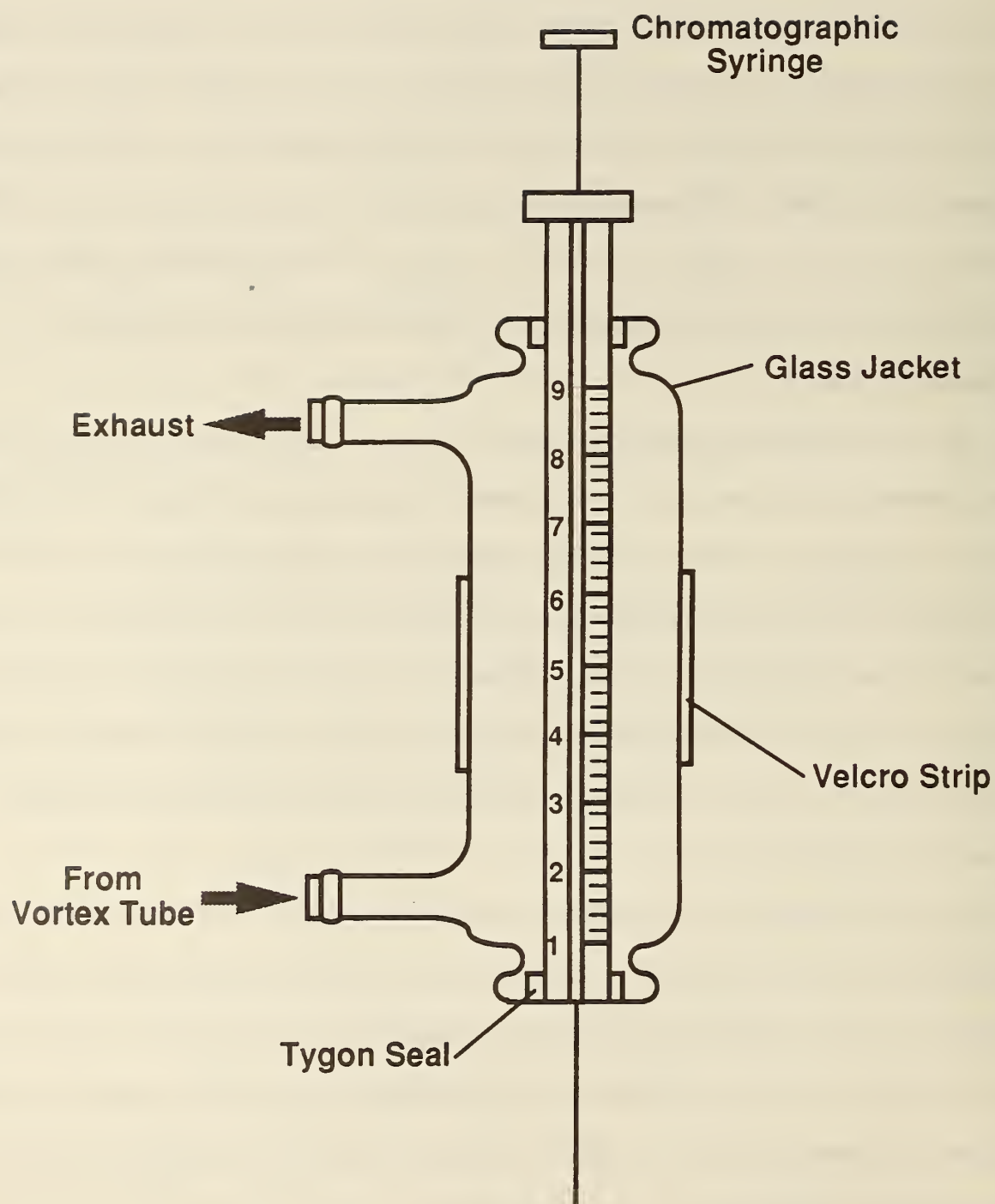
Figure 3. Vortex-cooled chromatographic syringe.

3.2 Analysis of Air

While capillary column gas chromatography is the most widely used separation method, gas analysis (and in some cases moisture analysis) is usually done using packed column methods. As was mentioned earlier, a great deal of the atmospheric trace analysis work has also involved the use of packed columns in conjunction with electron capture detection. Packed column methods have been applied extensively in the NIST protocol for the determination of air and moisture contamination of alternative refrigerant fluids. Air analysis is generally applied only to the more volatile samples that are gaseous under ambient conditions, or that are supplied as liquids under a vapor head space in a pressure cylinder.

The analysis of air in a fluid is easily done using a packed column (typically 2 m long, 0.6 m outside diameter), and a thermal conductivity detector. The carrier gas most often used is helium, although hydrogen* will provide somewhat higher sensitivity. Typical column flow rates are between 25 and 35 milliliters per minute. A stationary phase consisting of 80/100 mesh Porapak-QS can be used to resolve air from the refrigerant, although the resolution of oxygen and nitrogen will occur only at subambient temperatures [3]. The use of the more common molecular sieves are not generally recommended due to the strong tendency of many of the fluids to adsorb onto the surface. This adsorption needs to be investigated before such a material is used in a chromatographic analysis. The adsorption process is often highly exothermic, and sometimes enough heat is generated to decompose the refrigerant fluid. A recently developed molecular sieve, adsorption does not appear 4AXH7, is useful in a pelletized form (approximately 0.28 in diameter) in drying applications, since

---

* The use of hydrogen as a chromatographic carrier gas must involve strict adherence to applicable safety considerations.

adsorption does not appear to be a problem with the material. The chromatographic use of ground 4AXH7 is questionable, however, since in this form it appears to act more as a simple adsorbent rather than a typical molecular sieve material. There are a number of other porous polymer stationary phases that are of value in alternative refrigerant analyses. The most common materials are summarized in table 9 [3]. The retention indices W, X, Y, and Z are relative retention times for benzene, t-butanol, 2-butanone and acetonitrile, respectively. The choice will generally be determined by considerations of polarity and surface area.

Two methods of sample introduction can be used in air analysis: valve injection or the use of a gas-tight syringe. The use of the gas-tight syringe usually provides a precision level of no better than 5 percent, however this may be acceptable for rapid screening tests, especially when a valve system is not available. A thermostatted injection valve maintained at a temperature above approximately 150 °C (to prevent adsorption) provides much higher reproducibility. The sample loop usually has a volume on the order of 1 mL. Commercial chromatographic systems typically provide valve zone temperature control of ± 1 °C. The transfer lines leading from the sample cylinder to the sampling valve, which should all be passivated stainless steel, should also be heated using a steam trace or resistive element, again to prevent adsorption.

For many refrigerant analysis applications, the quantity of sample available for analysis is limited. An example of such a situation is in the analysis of a sample taken from a measurement apparatus such as a viscometer after measurements have been performed. In these cases, the gas injection apparatus shown in figure 4 is of value [11]. This apparatus is also useful in the analysis of fluids of a more hazardous nature, since not only the temperature but also the pressure can be measured and adjusted, and much less fluid needs to be handled. The heart of the injector is a manifold that incorporates a 10-port sampling valve having provisions for sample filling, venting and evacuation.

Table 9. Porous polymer packings[a]

POROUS POLYMER PHASES

Packing name	Max temp (°C)	Material type	Free-fall density (g/cm³)	Surface area (m²/g)	Pore diam av (μm)	Retention indices				Separation effected	Notes
						W	X	Y	Z		
Chromosorb 101	275	Styrene-divinylbenzene copolymer	0.30	<50	0.3—0.4	745	565	645	580	Free fatty acids, glycols, alcohols, alkanes, esters, aldehydes, ketones, ethers	Hydrophobic; condition at 250°C; not recommended for amines or anilines, lower retention times than obtained with Chromosorb 102
Chromosorb 102	250	Styrene-divinylbenzene copolymer	0.29	300—500	0.0085	650	525	570	460	Subambient temperature: H₂, O₂, N₂, Ar, NO, CO; ambient temperature: H₂, (air + NO + CO), CH₄, CO₂, H₂O, N₂O, C₂H₆; above ambient temperature: C₁—C₄ hydrocarbons, H₂S, COS, SO₂, esters, ethers, alcohols, ketones, aldehydes, glycols	May entrain some species; hydrophobic; condition at 225°C; not recommended for amines or nitriles; little tailing of water or oxygenated hydrocarbons
Chromosorb 103	275	Polystyrene cross-linked	0.32	15—25	0.3—0.4	720	575	640	565	Ammonia, light amines, light amides, alcohols, aldehydes, hydrazines	Hydrophobic; high affinity for basic species, not recommended for acidic species, glycols, nitriles, nitroalkanes
Chromosorb 104	250	Acrylontrile divinylbenzene copolymer	0.32	100—200	0.06—0.08	845	735	860	885	Sulfur gases, ammonia, nitrogen oxides, nitriles nitroalkanes, xylenols, water in benzene	Hydrophobic; condition at 225°C; not recommended for glycols and amines; moderately polar

a Taken from T.J. Bruno, P.D.N. Svoronos, Handbook of Basic Tables for Chemical Analysis, CRC Press, Boca Raton, 1989.

33

Table 9. Porous polymers packings (condt.)

Chromosorb 105	250	Acrylic ester (polyaromatic)	0.34	600—700	0.04—0.06	635	545	580	480	Permanent and light hydrocarbon gases; aqueous solutions of light organics such as formalin	Hydrophobic; less polar than Chromosorb 104; condition at 225 °C; Not recommended for acidic species, glycols, amines and amides
Chromosorb 106	250	Polystyrene cross-linked	0.28	700—800	0.05	605	505	540	405	Fatty acids from fatty alcohols, up to C_5; benzene from nonpolar organic compounds	Hydrophobic; not recommended for glycols and amines
Chromosorb 107	250	Acrylic ester cross-linked	0.30	400—500	0.8	660	620	650	550	Aqueous solutions of formaldehyde; alkynes from alkanes	Hydrophobic; moderately polar; not recommended for glycols and amines
Chromosorb 108	250	Acrylic ester cross-linked	0.30	100—200	0.25	710	645	675	605	Polar materials such as water, alcohols, aldehydes, glycols	Hydrophobic; condition at 250°C
Haysep A	165	Divinylbenzene/ethylene glycol dimethacrylate (high purity)	0.356	526	—	—·	—·	—·	—·	Separates permanent gases at ambient temperatures, and is useful for hydrocarbons to C_5, H_2S, H_2O at elevated temperatures	Relatively high polarity
Haysep B	190	Divinylbenzene/polyethyleneimine	0.330	608	—	—·	—·	—·	—·	Separates C_1 and C_2 amines, and trace levels of NH_3 and H_2O	High polarity
Haysep C	250	Divinylbenzene/acrylonitrile	0.322	442	—	—·	—·	—·	—·	Separates polar hydrocarbons; also HCN, NH_3, H_2S, H_2	Moderate polarity, with separation characteristics similar to Chromosorb 104
Haysep D	290	Divinylbenzene (high purity)	0.3311 (av)	795 (av)	0.0308—0351	—·	—·	—·	—·	Separates light gases; CO, CO_2, C_2H_2, C_2 hydrocarbons, H_2S, H_2O	Low-polarity polymer available in four formulations of different surface area (771—803 m²/g), density (0.3283—0.3834 g/ml), and porosity (64.2—70.4%)

34

Table 9. Porous polymers packings (condt.)

POROUS POLYMER PHASES (continued)

Packing name	Max temp (°C)	Material type	Free-fall density (g/cm³)	Surface area (m²/g)	Pore diam av (μm)	Retention indices				Separation effected	Notes
						W	X	Y	Z		
Haysep N	165	Divinylbenzene/ethylene glycol dimethacrylate (high purity)	0.355	405	—	—⁰	—⁰	—⁰	—⁰	Separation similar to Porapak materials; moderately high H_2O retention; see retention table	Low polarity polymer
Haysep P	250	Divinylbenzene/styrene	0.420	165	—	—⁰	—⁰	—⁰	—⁰	Separation of low molecular mass materials containing halogens, sulfur, water, aldehydes, ketones alcohols, esters, and fatty acids	Moderate to low polarity
Haysep Q	275	Divinylbenzene	0.351	582	—	—⁰	—⁰	—⁰	—⁰	Separation similar to Haysep P; see retention table	Low polarity
Haysep R	250	Divinylbenzene/N-vinyl-2-pyrollidinone	0.324	344	—	—⁰	—⁰	—⁰	—⁰	Separation similar to Haysep P; see retention table	Moderate polarity
Haysep S	250	Divinylbenzene/4-vinyl-pyridine	0.334	583	—	—⁰	—⁰	—⁰	—⁰	Separation similar to Haysep P; see retention table	Moderate polarity
Haysep T	165	Ethylene glycol dimethacrylate (high purity)	0.381	250	—	—⁰	—⁰	—⁰	—⁰	See retention table	High polarity
Porapak-Q	250	Ethylvinylbenzene-divinylbenzene copolymer	0.35	500—700	0.0075	630	538	580	450	Similar to Chromosorb 102	Similar to Chromosorb 102 condition at 250°C; most popular of all porous polymer phases

Table 9. Porous polymers packings (condt.)

Packing	Composition									Applications	Comments
Porapak-P	Styrene-divinylbenzene copolymer	250	0.28	100—200	—	765	560	650	590	Similar to Porapak-Q	Hydrophobic; low polarity; larger size than Porapak-Q, thus lower retention times are observed; not recommended for amines or anilines; condition at 250 °C
Porapak-N	Vinylpyrolidone	200	0.39	225—350	—	735	605	705	595	Similar to Chromosorb 105; high water retention; CO_2, NH_3, H_2O, C_2H_2, from light hydrocarbons	Condition at 175°C; not recommended for glycols, amines, or acidic species
Porapak-R	Vinylpyrolidone	250	0.33	300—450	0.0076	645	545	580	455	Ethers, esters, H_2O from chlorine gases (HCl, Cl_2), nitriles, nitroalkanes	Moderately polar; condition at 250°C; not recommended for glycols and amines
Porapak-S	Vinylpyridine	250	0.35	300—450	0.0076	645	550	575	465	Normal and branched alcohols, aldehydes, ketones, halocarbons	High polarity; not recommended for acidic species and amines; condition at 250°C
Porapak-T	Ethylene glycol dimethacrylate	200	0.44	250—300	0.009	—	675	700	635	Water in formalin (and other aqueous organic mixtures), retention characteristics similar to Chromosorb 107	Condition at 180°C; highest polarity of Porapak series; not recommended for glycols and amines
Porapak-QS	Ethylvinylbenzene-divinyl-benzene copolymer	250	—	—	—	625	525	565	445	Similar to Porapak-Q at lower operating temperatures, but useful for higher-molecular-weight solutes	Silanized Porapak-Q, reduces tailing of high-polarity compounds; condition at 250°C
Porapak-PS	Styrene-divinylbenzene copolymer	250	—	—	—	—	—	—	—	Similar to Porapak-P	Silanized Porapak-P, condition at 250°C.
Tenax-GC	p-2,6 Diphenylphenylene oxide polymer	375	0.37	18.6	—	—	—	—	—	Similar to Porapak-Q	Highest thermal stability of all porous polymers

• Retention indices are not available for these porous polymers, but a table of relative retentions on some representative solutes is included at the end of this section.

36

Table 9. Porous polymers packings (condt.)

Relative retention on some Haysep porous polymers

The following table provides relative retention values for Haysep polymers N, Q, R, S, T. These data were obtained using a 2-m long, 0.32-cm O.D. stainless steel column, using helium as the carrier gas.

HAYSEP POLYMER

| Compound	N	Q	R	S	T
Hydrogen	0.19	0.143	0.17	0.19	0.21
Air	0.23	0.186	0.2	0.21	0.25
Nitric oxide	0.25	0.217	0.21	0.23	0.33
Methane	0.30	0.256	0.28	0.3	0.35
Carbon dioxide	0.71	1.15	0.50	0.52	0.85
Nitrous oxide	0.80	1.43	0.59	0.59	—
Ethylene	0.83	0.74	0.78	0.78	0.9
Acetylene	1.41	0.74	1.0	0.87	2.11
Ethane	1.0	1.0	1.0	1.0	1.0
Water	10.1	1.45	0.68	4.12	19.1
Hydrogen sulfide	2.1	1.40	1.73	1.87	2.88
Hydrogen cyanide	1.93	2.31	15.6	8.26	28.8
Carbonyl sulfide	2.82	2.33	2.46	2.63	3.4
Sulfur dioxide	12.0	3.05	9.78	17.8	19.0
Proplyene	4.66	3.20	3.45	3.65	4.91
Propane	4.66	3.67	3.88	4.1	4.63
Propadiene	6.50	4.12	4.39	4.7	7.55
Methylacetylene	9.5	4.12	4.84	5.14	11.3
Methyl chloride	7.43	3.93	4.67	4.92	9.2
Vinyl chloride	14.9	6.04	9.04	9.7	17.3
Ethylene oxide	17.7	6.06	8.78	9.7	23.3
Ethyl chloride	35.0	12.25	19.3	20.7	43.2
Carbon disulfide	—	32.4	—	—	40.7
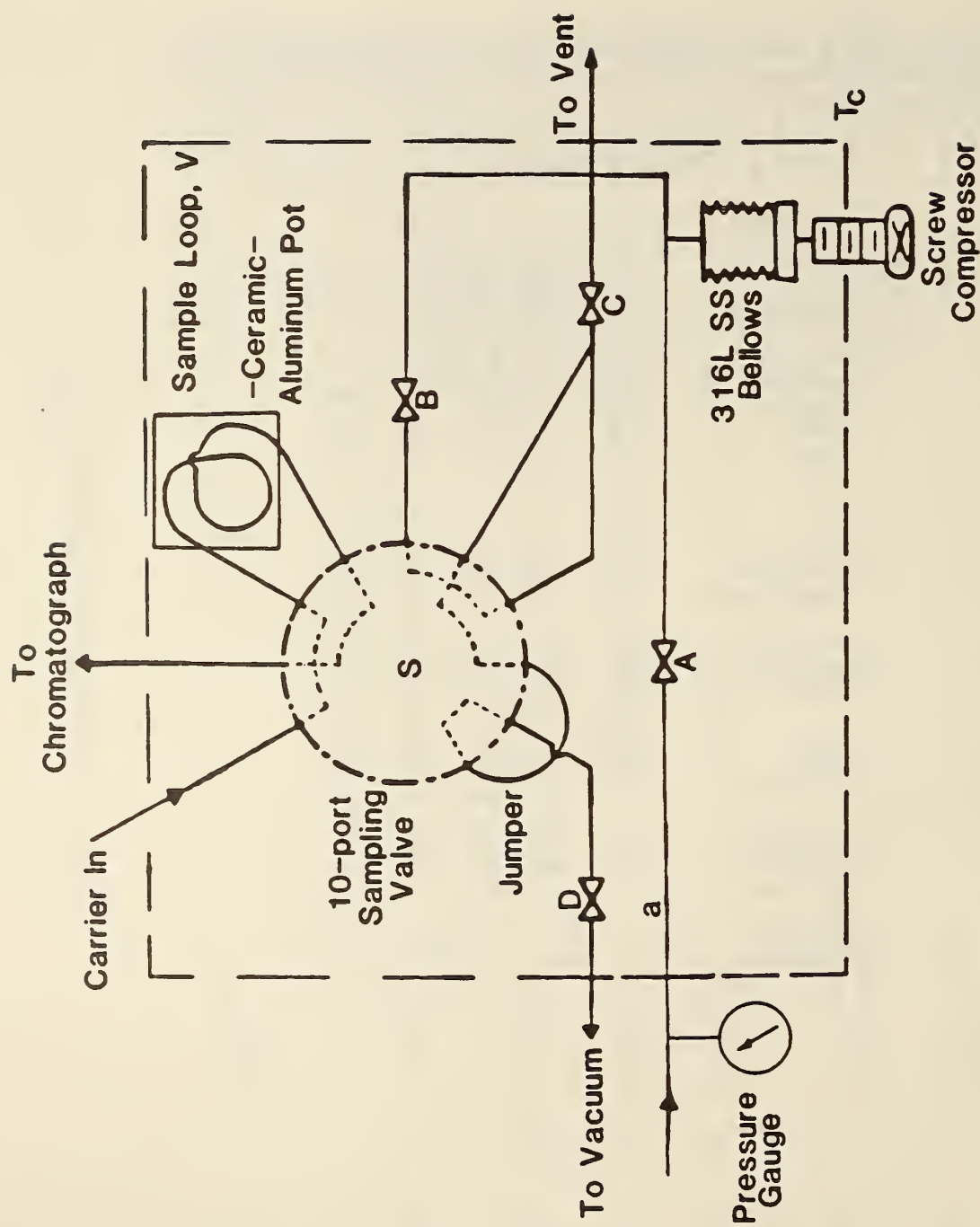
Figure 4. Gas sampling and injection system for small or hazardous samples.

The sample is added through valve A, and is held between valves A and B in a flexible stainless steel bellows that has a volume of approximately 15 mL. The bellows can be compressed or relaxed using a hand screw, thus allowing the pressure of the fluid to be adjusted. Valve B allows the introduction of sample into the sampling valve, which is equipped with a 0.05 mL sample loop. The loop itself is potted in a silicon dioxide/aluminum ceramic capsule specially formulated for its high thermal conductivity and thermal mass. Valve C allows the fluid to be released into a vent line (connected to a fume hood).

The entire apparatus is mounted within a relatively massive (1050 cm^3) aluminum block, which is thermostatted (to within ± 0.03 °C) using a set of 5 proportionally controlled cartridge heaters. The aluminum block, mounted in an insulated box, serves to integrate out large temperature variations that may be present among the various components of the apparatus. A thermocouple is potted with the sample loop to provide a direct temperature reading of the sample using a digital voltmeter. This is useful to allow sufficient time for thermal equilibration of the sample, and to ensure that the sample size is reproducible. The pressure of the sample inside the loop is measured as well, and the chromatographic area counts are usually corrected to atmospheric pressure. The pressure can be determined using an in-situ strain gauge transducer, or more commonly by injecting with the loop at atmospheric pressure, the actual value of which is read from an electronic barometer. The control and measurement of both temperature and pressure allows the injection of a constant quantity of sample into the column.
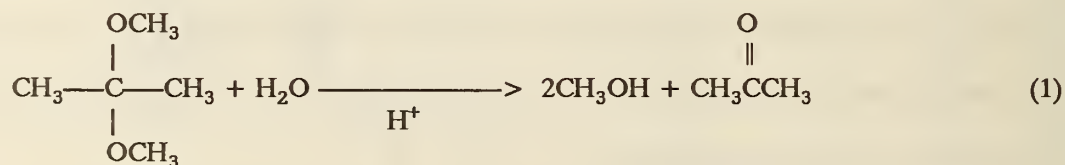
The reproducibility of the chromatographic area counts obtained using this injection apparatus is far better than that which can be obtained using a simple commercial gas sampling system. When dealing with the small samples that this device was constructed for, the improvement in reproducibility is usually better than an order of magnitude.

3.3 Analysis of Water

The analysis of water present at low levels in alternative refrigerants may be performed using two principal techniques, some adaptation of a chromatographic method or the Karl Fischer titration. A number of chromatographic methods were developed in the early sixties, many of which involved the conversion of the water to another compound more amenable to sensitive detection. The more important reactive methods will be summarized here, since they can be of great use in many laboratories.

A sample (gaseous or liquid) containing traces of water can be passed through a reactor containing ground calcium carbide [12]. The water is converted to acetylene, which can then be detected using the highly sensitive flame ionization detector. This approach has been used for samples containing water at levels as low as 3 ppm (by mass), and typically has a reproducibility of between 7 and 15 percent.

Another derivatization method which can be applied mainly to liquid samples involves the conversion of water to acetone and methanol using 2,2-dimethoxypropane [13] in the presence of methane sulfonic acid (catalyst) according to:

$$CH_3 \overset{\displaystyle OCH_3}{\underset{\displaystyle OCH_3}{\overset{|}{\underset{|}{C}}}} CH_3 + H_2O \xrightarrow[H^+]{} 2CH_3OH + CH_3\overset{\displaystyle O}{\overset{\|}{C}}CH_3 \qquad (1)$$

Flame ionization detection can be used to detect either product, although the methanol can produce tailing peaks on many columns that may be difficult to accurately quantitate. The reproducibility with liquid samples is in the range of 10 percent. To apply this approach to gaseous samples, a pressure vessel shaker (figure 5) has been constructed and used with mixed results. The

40
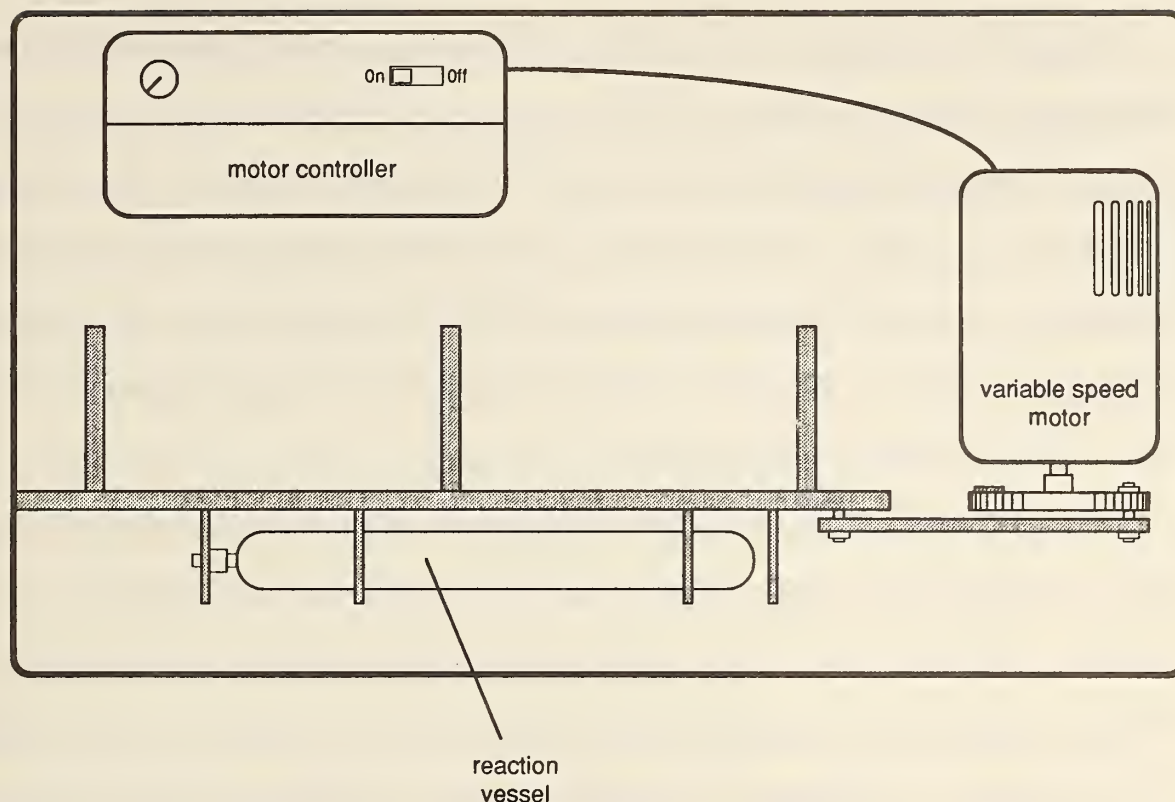
Figure 5. Pressure vessel/reactor shaker.

main difficulty is the loss of product upon decompression. Typical reproducibility with gases is in the range of 15 and 20 percent.

The reaction of water with sodium aluminum hydride ($NaA\ell H_4$) to produce hydrogen can be used in conjunction with thermal conductivity detection, when nitrogen or argon is used as the carrier gas. The $NaA\ell H_4$ is held in a reactor dissolved in diethyleneglycol dimethyl ether. The separation of hydrogen from the carryover solvent and refrigerant fluid is usually trivial, since hydrogen is poorly retained and usually emerges from the column quickly. The possibility of obtaining negative peaks (due to the low thermal conductivity carrier gas) can sometimes complicate the analysis. It is therefore best to use nitrogen or argon as the carrier gas. This approach is reported to be useful down to 2 to 3 ppm by mass.

41

Since the development of more sophisticated thermal conductivity detectors (such as constant current bridges and single filament cells), the analysis of water directly after separation on an appropriate column is relatively easy [15,16]. Porapak-QS provides very good resolution between the water peak and that of the refrigerant fluid(s). A typical analysis usually takes no more than thirty minutes for a 1 mL sample at atmospheric pressure. Water concentrations as low as 0.5 ppm by mass can be detected. The reproducibility is usually on the order of a few percent, provided the injection valve and transfer lines are carefully heated.

The Karl Fischer titration is one of the most important techniques for the analysis of moisture [17]. The method is very sensitive (down to the 1 ppm level by mass) and reproducible to within 1 percent. The classical Karl Fischer reagent consists of iodine, pyridene, and sulfur dioxide, in methanol or ethylene glycol monomethyl ether as a solvent. The reagent must be freshly prepared for each analysis. The addition of the sample containing water initiates the following two-step sequence of reactions (with methanol solvent):

$$
C_5H_5N \cdot I_2 + C_5H_5N \cdot SO_2 + C_5H_5N + H_2O \longrightarrow
$$

$$
2\, C_5H_5NH^+I^- + C_5H_5N^+SO_3^- \tag{2}
$$

$$
C_5H_5N^+SO_3^- + CH_3OH \longrightarrow C_5H_5NH^+CH_3OSO_3^-
$$
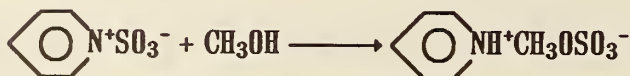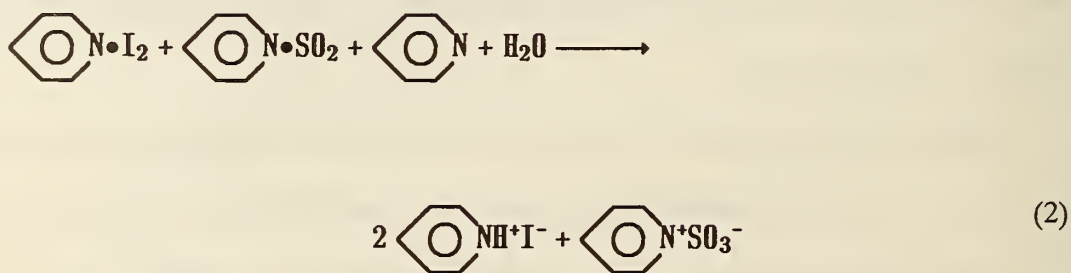
The alcoholic solvent is needed to drive both reactions to completion. Because of the difficulty in preparing and storing the classical reagent, most analyses are now done using a commercial proprietary formulation that is free of pyridene. The end point of the titration is usually detected bipotentiometrically, by noting the abrupt change in potential between two platinum electrodes (the dead-stop method). When I_2 and I^- are present, a constant current can flow between the electrodes at a very low applied voltage. At the end point, the current flow must occur mainly through the reduction of the methanol solvent to formate ion and elemental hydrogen. Most pH meters are equipped with an additional pair of receptacles labeled K-F, that are used for this measurement. The change in potential associated with the onset of this new electrode process is usually several orders of magnitude.

A more convenient approach to the Karl Fischer titration is the use of coulometrically generated iodine, usually obtained from dissolved potassium iodide in a pyridene-free amine solvent. A coulometric apparatus has two sets of electrodes. One set of electrodes, usually in the form of platinum screens separated by a glass frit, generates the iodine by anodic oxidation. The sensing electrodes consist of dual platinum probes. The end point can be detected using the dead stop method as described above, or more commonly using an amperometric method. In this method, the instrument measures the sharp decrease in resistance that accompanies the first trace of excess I^- ion which appears after the elimination of the water (from the sample) in the cell. This technique has a sensitivity in the 1 ppm range (by mass), and a reproducibility of approximately 0.5 percent.

The coulometric Karl Fisher titration can be used with both liquid and gaseous refrigerant fluids. Liquids are simply injected into the cell using a liquid chromatography syringe, and the weight of the sample is determined by difference. For gaseous samples, a small transfer line terminating in a fritted diffuser is used for sample introduction. With both gases and liquids, the cell is vigorously stirred during sampling and titration.

4. REACTION SCREENING

Many of the fluids being considered as alternative refrigerants are much more chemically reactive than the older fully halogenated methanes. Most of them are incompatible with alkali and alkaline earth metals, and finely divided metals such as aluminum and beryllium. Some will undergo slow hydrolysis reactions in the presence of water, and several of ethylene-based fluids react violently in the presence of air and moisture. While most of the highly reactive chlorofluorobromoethanes and -ethylenes are not under consideration as alternative refrigerants, these compounds can be inadvertently synthesized in an experimental apparatus. Almost all of the fluids being studied will undergo thermal decomposition to form highly corrosive acid anhydrides and highly toxic carbonyl halides and carbon monoxide. In apparatus having optical access, the formation of HF is particularly dangerous, since optical cell windows are quickly etched and rendered useless. It is therefore very important to assess the potential for reaction or decomposition in advance of thermophysical property measurement. There are four main causes of fluid decomposition or reaction: the chemical nature of the fluid (enthalpy and entropy of formation, for example), the conditions (mainly temperature and pressure) prevailing in the measurement apparatus, the construction materials of the apparatus, and the residence time of the fluid in the apparatus.

The reactivity of a given fluid is studied using the reaction screening apparatus shown in figure 6 [18]. The heart of the apparatus is a small, thick-walled pressure vessel machined from a section of 316L stainless steel (AISI designation) bar stock. The vessel has a sample well that holds the test fluid under conditions of temperature and pressure and residence time, to simulate the conditions that exist in the measurement apparatus. Alternatively, a glass, quartz or Teflon liner can be placed in the well to contain the sample. This vessel, which has a bolted gasket closure, can safely contain a pressure of 130 MPa at a temperature of 250 °C. Tolerances and specifications for this vessel far exceed the requirements set forth by the ASME [19]. The top of the vessel accommodates welded

feed-throughs used for the introduction of test fluid and the removal of samples for analysis. Sealing of the vessel is provided by either a 316L stainless steel gasket or a gasket machined from 25 percent glass-filled Teflon. A thermowell in the vessel accommodates a thermocouple which, when referenced to a known temperature, provides the vessel temperature using a digital voltmeter (DVM in figure 6). The vessel is protected with a safety head containing a rupture disk set for 68 MPa.

The pressure vessel rests within an aluminum jacket containing cartridge heaters (not shown in figure 6), thermally decoupled from the jacket by an air space. The heaters are under proportional control, and provide temperature control to within $\pm$ 0.05 °C. The aluminum jacket provides thermal uniformity by integrating out temperature gradients. The vessel and jacket are contained in an air-tight convection oven (represented by T_c in the figure). The oven is of welded steel insulated with bonded mineral wool board, and is maintained under a slight negative pressure. An inert gas purge line (usually the boil-off from a liquid nitrogen Dewar) is available to use with potentially explosive fluids.

Sampling of the contents of the pressure vessel can be done several ways during the course of a reaction screening test. The sample line is directly connected to a chromatographic sampling valve. This is most convenient for the analysis of gaseous or highly volatile products using the packed column methods discussed earlier. An external sampling cylinder can also be used to collect the fluid. Collection is usually done by cooling the cylinder in liquid nitrogen and condensing the fluid inside, while the pressure vessel is hot. Of course, at the end of a test period, the oven can be cooled (by simply increasing the flowrate of the liquid nitrogen boil-off purge line), and the chilled pressure vessel removed from the aluminum jacket. Any remaining liquid sample can then be collected.
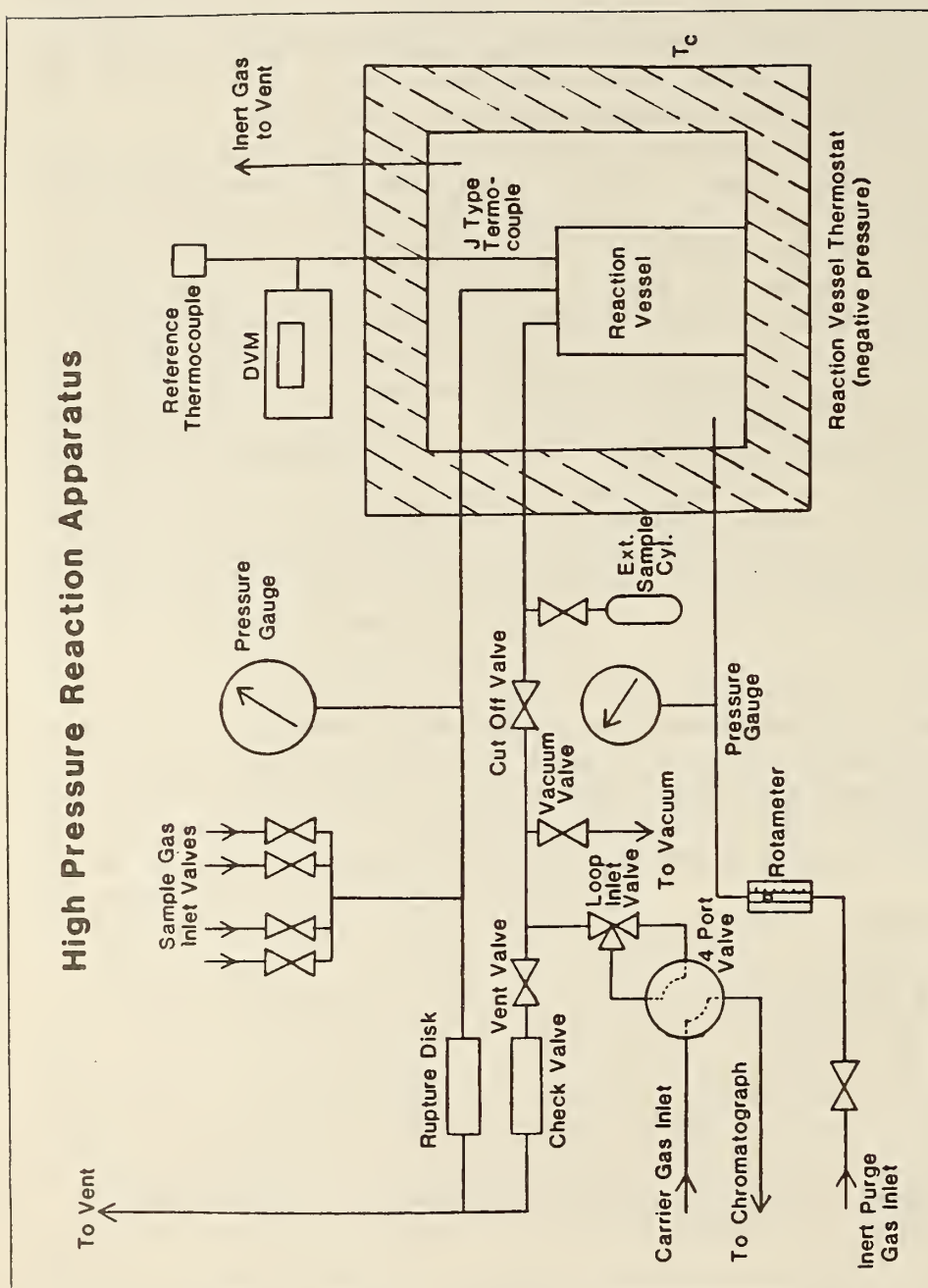
Figure 6. Reaction screening apparatus.

46

The catalytic effects of many common apparatus construction materials can be assessed using this apparatus. Many of these materials, such as the stainless steels, copper, glasses, etc., can be obtained as fine powders of known and controllable surface area. A test fluid can be held inside the pressure vessel in contact with an appropriate amount of powdered material to simulate the worst case conditions. As before, analysis follows a preselected residence time under the chosen temperature and pressure.

The catalytic effects of materials can also be studied using reaction chromatography [20]. In this technique, a bed of granular or powdered material (such as powdered stainless steel) is contained in a short packed precolumn placed directly ahead of a packed analytical column using zero-dead-volume fittings. The precolumn is typically 3 - 5 cm in length. Both of the columns are coiled in the oven of a gas chromatograph, although, if desired, the precolumn can be independently heated using an insulated block containing a cartridge heater, or by wrapping it with heating tape. The sample is then applied using a syringe (at room temperature or cooled to -40 °C, as discussed earlier) or a sampling valve. The reaction products which form are then focused on the head of the analytical column which follows the precolumn. Since this is a packed column method, the usual choice of detector will be thermal conductivity or electron capture. A mass selective detector can be used if interfaced through a molecular jet separator, although this will cause a loss of resolution [1].

There are a number of drawbacks to the reaction chromatography method, although it can be used to quickly provide an idea of the decomposition likely to be encountered in a measurement. The chromatographic column is operated at a relatively low pressure, and the sample is diluted with carrier gas. In addition, the ratio of the sample to precolumn surface area is essentially uncontrollable. It is therefore unreasonable to expect duplication of the conditions which will be present in, for example, a P-V-T apparatus. As the temperature of the precolumn is changed, the carrier gas flowrate will change and thus require adjustment. From an analytical point of view, it is inconvenient to use such a precolumn with capillary chromatographic columns.

5. ACKNOWLEDGMENT

6. REFERENCES

[1] Bruno, T.J. Chromatographic and Electrophoretic Methods, Prentice-Hall, Englewood Cliffs, N.J., 1990.

[2] Bruno, T.J. Spectroscopic Library for Alternative Refrigerant Analysis, National Institute of Standards and Technology Special Publication, 1990.

[3] Bruno, T.J., Svoronos, P.D.N., CRC Handbook of Basic Tables for Chemical Analysis, CRC Press, Boca Raton, FL, 1989.

[4] McLafferty, F.W. Interpretation of Mass Spectra, 3rd ed., University Science Books, Mill Valley, CA, 1980.

[5] Chapman, J.R. Practical Organic Mass Spectrometry, John Wiley and Sons, Chichester, 1985.

[6] Rose, M.E., Johnstone, R.A.W., Mass Spectrometry for Chemists and Biochemists, Cambridge University Press, Cambridge, MA, 1982.

[7] George, W., McIntyre, P., Infrared Spectroscopy, John Wiley and Sons (published on behalf of ACOL) Chichester, 1987.

[8] Colthup, N.B., Daly, L.H., Wiberley, S.E., Introduction to Infrared and Raman Spectroscopy, Academic Press, New York, 1975.

[9] Bruno, T.J., Anal. Chem., 58, 1596, 1986.

[10] Bruno, T.J., Vortex Cooled Syringe for Chromatography, U.S. Patent Appl. Ser. No. 07/42879.

[11] Bruno, T.J., J. Chromatogr. Sci., 23, 325, 1985.

[12] Knight, H.S., Weiss, F.T., Anal. Chem., 34, 749, 1962.

[13] Hager, M., Baker, G., Proc. Mont. Acad. Sci., 22, 3, 1962.

[14] Berezkin, V.G., Mysak, A.Ye., Polak, L.S., Neftekhimiya, (Trans. Sel. Art. Petroleum Chem., USSR), 4, 156, 1964.

[15] Swensen, R.F., Keyworth, D.A., Mat. Res. Stds., pg. 525, Dec. 1967.

[16] Hollis, O.L., Haynes, W.V., J. Gas Chromatogr., 4, 235, 1966.

[17] Harris, D.C., Quantitative Chemical Analysis, 2nd ed., W.H. Freeman and Co., NY, 1987.

[18] Bruno, T.J., Hume, G.L., J. Res Nat. Bur. Stds. (U.S.), 90, 255, 1985.

[19] ASME Boiler and Pressure Vessel Code, Sec. VIII: Unfired Pressure Vessels, NY, American Society of Mechanical Engineers, 1965.

[20] Bruno, T.J., Straty, G.C., J. Res. Nat. Bur. Stds., (U.S.) 91, 135, 1986.

Appendix 1.

Sources of toxicology and physical properties

1. Toxic and Hazardous Industrial Chemicals Safety Manual, The International Technical Information Institute, Tokyo, Japan, 1975.

2. Matheson Gas Data Book, The Matheson Company, East Rutherford.

3. Sax, N.I., Dangerous Properties of Industrial Materials, 6th ed., Van Nostrand Reinhold, New York, 1984.

4. Verschueren, K., Handbook of Environmental Data on Organic Chemicals, Van Nostrand Reinhold Co., New York, 1977.

5. Weast, R., CRC Handbook of Chemistry and Physics, 69th ed., CRC Press, Boca Raton, 1988.

6. Sax, N.I., Cancer Causing Chemicals, Van Nostrand Reinhold, New York, 1981.

7. The Condensed Chemical Dictionary, 8th ed., Van Nostrand Reinhold Co., New York, 1966.

8. Threshold Limit Values and Biological Exposure Indices for 1988-1989, American Conference of Governmental Industrial Hygienists, Cincinnati, 1988.

9. Dorland's Illustrated Medical Dictionary, 28th ed., W.B. Saunders Co., Philadelphia, 1989.

10. G. Weiss, ed., Hazardous Chemicals Data Book, 2nd. ed., Noyes Data Corp., Park Ridge, N.J., 1986.

11. The Merck Index, Merck and Co., Inc., Rahway, 1983.

The following DIALOG databases are useful to obtain toxicological and physical properties.

> Chemname
> Chemical Abstracts
> Occupational Safety and Health
> Registry of Toxic Effects of Chemical Compounds

The following BRS database are useful to obtain toxicological and physical properties.

Appendix 2

A computer Program* to aid in the interpretation of mass spectra, especially in the analysis of chlorofluoroethanes and -ethylenes.

* Written in Turbo-Pascal, 5.0, Borland International, Scotts Valley, CA.

```
program spectrometry;
{
    finished 5/25/89
    revised 2-4/90
    N.I.S.T. Thermophysics Division
    325 S. Broadway
    Boulder, Co 80303
    Tom Bruno Ph.D.
    mail code 584.03

    This program is comprised of three different routines that cam be
    used separately or in conjunction with one another.
}


{
    This program is designed to assist the user in the interpretation
    of mass spectra.  It is especially useful in work involving chloro-
    fluoro-bromo ethanes and ethylenes.

    Part 1)   The program uses various rules about mass spectra to
              determine possible molecular formulae of an ion.

    Part 2)   The program uses the relative atomic mass of an ion to
              determine possible formulae using basic combinatoric rules.

    Part 3)   The program searches through a database of know ions
              given a m/e value.
}
  uses crt;


{  declare global constants and types }

    const
        MAXSIZE=17;
        MAXELE=108;
        MAXCOMB=200;
        maxpsize = 7;

    type
        sttype = string[maxsize];
        stttype = string[2];
        line = string[80];
        page = array[1..23] of line;
        coefflist = array[1..MAXSIZE] of integer;
        dualarray = array[1..2] of integer;
        double = array[1..MAXSIZE] of dualarray;

        datalist = record
                n:sttype;
                s:stttype;
                w:real;
            end;
        largearray =array[1..MAXELE] of datalist;
        smallarray = array[1..MAXSIZE] of datalist;
        combolist = array[1..MAXCOMB] of coefflist;
        elementlist = largearray;
        ionlist = smallarray;
    pnode = ^node;
    node = record
            data:real;
            next:pnode;
          end;
    table = array[0..7,0..4] of pnode;
    element = record
```

```pascal
                    name:string[20];
                    symbol:string[2];
                    weight:real;
                  end;
     result_record = record
                    nitrogens:boolean;
                    nitrogensmax:integer;
                    carbons:integer;
                    carbonscalc:integer;
                    sulfurs:integer;
                    sulfurscalc:integer;
                    chlorines:integer;
                    bromines:integer;
                    oxygens:integer;
                    hydrogens:integer;
                  end;
     fragment_record = record
                    number:integer;
                    name:string[12];
                    intensity:string[20];
                  end;

     element_list = array[0..6] of element;
     fnamestring = string[12];
     intensity_array = array[1..7] of real;


{ program_description - print out a title paragraph }

procedure program_description;
begin
   clrscr;
   writeln('                         Mass Spectromety Analysis');
   writeln('                         ------------------------');
   writeln;
   writeln('     This package will assist the user in the interpretation of ',
          'mass ');
   writeln;
   writeln(' spectra, for Qualitative analysis and Structure determination.');
   writeln;
   writeln(' PRESS <RETURN> TO START PROGRAM');
   readln;
end;



{***********************************************************************}
{ BEGINNING OF PART 1 }


{ open_file - opens a file to the beginning of the file }

procedure open_file(var myfile:text;fname:fnamestring);

   begin
     assign(myfile,fname);
     reset(myfile);
   end;


{ main1_description - prints description of part1 }

procedure main1_description;
   begin
     clrscr;
     writeln('                         Molecular Formula Analysis');
```

```
writeln('                              --------------------------');
writeln;
writeln;
writeln(' The following routines are used in the mass spectroscopic ',
        'determination of the');
writeln;
writeln(' the molecular formula of an organic compound.  These routines '
        ,'should be ');
writeln;
writeln(' applied to the molecular ion peak and its isotopic cluster.',
        '  The molecular');
writeln;
writeln(' ion (if it is present) should be the highest mass in the ',
        'spectrum and must be');
writeln;
writeln(' an odd-electron ion and must be capable of yielding all other',
        ' important ions ');
writeln;
writeln(' of the spectrum via a logical neutral species loss.  The ',
        'elements that are ');
writeln;
writeln(' assumed to possibly be present are carbon, hydrogen, ',
        'nitrogen, the halogens,');
writeln;
writeln(' sulfur, and/or oxygen.  The molecular formula that can be ',
        'derived is not the');
writeln;
writeln(' only possible one and consequently help from nuclear magnetic',
        ' resonance');
writeln;
writeln(' spectrometry and infrared spectrophotometric data maybe',
        ' necessary for the');
writeln;
write(' PRESS <RETURN> TO CONTINUE');
readln;
clrscr;
writeln(' final determination.');
writeln;
writeln(' The routines used in the analysis use the following rules to',
        ' determine the');
writeln;
writeln(' molecular formula.');
writeln;
writeln(' RULES:');
writeln;
writeln(' 1)  An odd molecular ion value suggests the presence of an odd'
        ,' number of');
writeln('      nitrogens.  An even molecular ion is due to the presence',
        ' of zero or an');
writeln('      even number of nitrogens');
writeln;
writeln(' 2)  The maximum number of carbons can be calculated from the',
        ' formula');
writeln;
writeln('         Relative intensity of M+1      100');
writeln('         ------------------------   x  ---');
writeln('         Relative intensity of M+      1.1');
writeln;
writeln('     where M+1 is the peak next to that of the molecular ',
        'ion(M+).  This');
writeln('     rule gives the maximum number of carbons, but not ',
        'necessarily the right');
writeln('     number.');
writeln;
writeln(' 3)  The maximum number of sulfurs can be calculated from the ',
        'formula ');
```

```
        writeln;
        write(' PRESS <RETURN> TO CONTINUE');
        readln;
        clrscr;
        writeln('        Relative intensity of M+2   x  100');
        writeln('        ------------------------       ---');
        writeln('        Relative intensity of M+       4.4');
        writeln;
        writeln('     where M+2 is the peak whose intensity corresponds to the',
                ' mass M+2.');
        writeln;
        writeln(' 4)  The number of chlorines and/or bromines can be derived ',
                ' from the table');
        writeln('     showing the relative ratio of the isotope cluster of ',
                ' the molecular ion.');
        writeln;
        writeln(' 5)  The difference should be only oxygen and hydrogen.  These',
                ' rules assume');
        writeln('     absence of phosphorous, silicon, or any other elements.');
        writeln;
        writeln(' The user is prompted for the mass (M) of the parent ion, and',
                ' the relative');
        writeln;
        writeln(' intensities of the peaks adjacent to M.  The user is also ',
                'prompted for the');
        writeln;
        writeln(' absolute intensities of P+2, P+4, etc, to determine the ',
                'number of Chlorines');
        writeln;
        writeln(' and/or Bromines in the ion.');
        writeln;
        write(' PRESS <RETURN> TO GO TO SECTION MENU ');
        readln;
        clrscr;
    end;

{ table_init - initializes the BrCl array of linked lists to nil }


procedure table_init(var BrCl_table:table);

var
  i,j:integer;

    begin
      for i:=0 to 7 do
        for j:=0 to 4 do
          BrCl_table[i,j]:=nil;
    end;

{ read_tabledata - read in the bromium and chloride data into the BrCl
                   data structure.   }


procedure read_tabledata(var myfile:text; var BrCl_table:table);

var
    i,j,k:integer;
    temp:pnode;

    begin
      while (not eof(myfile)) do
      begin
      readln(myfile,i,j);
      k:=1;
      while (k<=(i+j)) do
```

```pascal
          begin
            if (k=1) then
              begin
                new(BrCl_table[i,j]);
                new(BrCl_table[i,j]^.next);
                temp:=BrCl_table[i,j]^.next;
              end
            else
              begin
                new(temp^.next);
                temp:=temp^.next;
              end;
            readln(myfile,temp^.data);
            temp^.next:=nil;
            k:=k+1;
          end;
          end;
      end;


{ table_input - open the file for data file for the Bromium Chloride
                data and store it in the Bromium Chloride data structure }

procedure table_input(var BrCl_table:table);

var
    myfile:text;

    begin
      open_file(myfile,'table.dat');
      read_tabledata(myfile,BrCl_table);
      close(myfile);
    end;

{ element_input - Read in the elements for use in determining molecular
                  formula from rules }

procedure element_input(var element_data:element_list);

var
  i:integer;
  myfile:text;

begin
    open_file(myfile,'element.dat');
    while (not eof(myfile)) do
    for i:=0 to 6 do
      begin
        readln(myfile,element_data[i].name);
        readln(myfile,element_data[i].symbol);
        readln(myfile,element_data[i].weight);
      end;
end;

{ load_data - loads data for part 1 }

procedure load_data(var BrCl_table:table; var element_data:element_list);

    begin
      clrscr;
      writeln('loading data from disk');
      writeln('please be sure the program disk is in the default drive');
      writeln('PRESS <RETURN> WHEN THE DISK IS IN THE DRIVE');
      readln;
      table_init(BrCl_table);
      table_input(BrCl_table);
```

```
        element_input(element_data);
    end;

{ input_data - ask for various inputs necessary for part 1 }

procedure input_data(var m,m1,m2,percent:real;var p_intensity:intensity_array;
                     var me,psize:integer);

var
    intensity,temp2:real;
    ist:line;
    test,subtract,select:boolean;
    temp,i:integer;
    answer:char;

    begin
     repeat
       psize:=0;
       ist:='';
       test:=false;
       write('Please input the parent ion mass, M+ :  ');
       readln(me);
       write('Please input the absolute intensity of M+:  ');
       readln(m);
       write('Please input the absolute intensity of M + 1:  ');
       readln(m1);
       m1:=(m1/m)*100;
       writeln;
       writeln('To quit adding the absolute intensities, enter anything except',
               ' a number.');
       writeln;
       while ((not test) and (psize<maxpsize)) do
         begin
           write('Please input the absolute intensity of M + ',(2*(psize+1)),':   ')
           readln(ist);
           temp:=ord(ist[1]);
           i:=1;
           temp2:=1.0;
           if (temp=45) then
               begin
                 subtract:=true;
                 i:=i+1;
                 temp:=ord(ist[i]);
               end
           else
               subtract:=false;
           intensity:=0;
           if ((temp=46) or ((temp<58) and (temp>47))) then
             begin
               select:=true;
               while (((temp=46) or ((temp<58) and (temp>47))) and (i<80))  do
                 begin
                   if (temp=46) then
                     select:=false
                   else
                     if select then
                       begin
                         if (subtract) then
                           intensity:=(10*intensity)-(temp-48)
                         else
                           intensity:=(10*intensity)+(temp-48)
                       end
                     else
                       begin
                         temp2:=temp2*0.1;
                         if (subtract) then
```

59

```pascal
                          intensity:=intensity-((temp-48)*temp2)
                      else
                          intensity:=intensity+((temp-48)*temp2);
                      end;
                    i:=i+1;
                     temp:=ord(ist[i]);
                    end;
               end
            else
               test:=true;
            if (not test) then
                begin
                  psize:=psize+1;
                  p_intensity[psize]:=intensity;
                end;
            for i:=1 to 80 do
               ist[i]:=' ';
          end;
        if (psize=0) then
           percent:=0
        else
          repeat
            writeln('Please input a percent error for calculating the number of',
                     ' Bromine and');
            writeln('Chlorines.   (note that a choice of 0 percent is not ',
                     'possible due to');
            write(' rounding errors in the computer) :   ');
            readln(percent);
            if (percent<0.0) then
               writeln('Percentage must be positive.   Try again.');
            until (percent>=0.0);
        m2:=(p_intensity[1]/m)*100;
        repeat
        write('Is this data Ok (y/n)?   ');
        readln(answer);
        answer:=upcase(answer);
        until answer IN['Y','N'];
        until (answer='Y');
      end;

{ output_results - outputs results of Part 1 calculations }

procedure output_results(var result:result_record;me:integer);
    begin
        writeln('                                       RESULTS');
        writeln('                                       -------');
        writeln;
        if (result.nitrogens) then
            writeln('The ion suggests zero or an even number of nitrogens ',
                     'present.')
        else
            writeln('The ion suggests an odd number of nitrogens present.');
        writeln('The maximum number of nitrogen atom(s) as determined by the ',
                 'ion mass = ',result.nitrogensmax);
        writeln;
        writeln('The ion has at most ',result.carbons,' carbon atom(s).');
        if (result.carbonscalc<result.carbons) then
         begin
          writeln('          Since the ion has mass ',me,', the maximum number of ',
                   ' carbons');
           writeln('          possible is ',result.carbonscalc,'.   The number shown',
                   ' above is');
           writeln('          the result of rule number 2 (see description)');
         end;
        writeln;
        writeln('The ion has at most ',result.sulfurs,' sulfur atom(s).');
```

```pascal
        if (result.sulfurscalc<result.sulfurs) then
          begin
            writeln('          Since the ion has mass ',me,', the maximum number of',
                    ' sulfurs');
            writeln('          possible is ',result.sulfurscalc,'.  The number shown',
                    ' above is');
            writeln('          the result of rule number 3 (see description)');
          end;
        writeln;
        writeln('The ion has ',result.chlorines,' chlorine atom(s).');
        writeln('The ion has ',result.bromines,' bromine atom(s).');
        writeln;
        write(' PRESS <RETURN> TO CONTINUE');
        readln;
    end;


{ calculate - performs calculations for Part 1 with various
              rules about mass spectra. A summary can be found in:
              Bruno, T.J., Svoronos, P.D.N, CRC Handbook of Basic Tables
              for Chemical Analysis, CRC Press, Boca Raton, 1989.}

procedure calculate(m,m1,m2,percent:real; me,psize:integer; var p_intensity:
                    intensity_array; var element_data:element_list;
                    var result:result_record; var BrCl_table:table);
var
    i,j,k:integer;
    equal:boolean;
    temp:pnode;

    begin
      writeln('starting calculations');
      if ((me mod 2) = 0)
          then
            result.nitrogens:=true
          else
            result.nitrogens:=false;
      result.nitrogensmax:=trunc(me/element_data[2].weight);
      if (((result.nitrogensmax mod 2)=0) and (not result.nitrogens) or
          ((result.nitrogensmax mod 2)=1) and (result.nitrogens)) then
            result.nitrogensmax:=result.nitrogensmax-1;
      result.carbons:=trunc((m1/1.1)+0.5);
      result.carbonscalc:=trunc(me/element_data[1].weight);
      result.sulfurs:=trunc((m2/4.4)+0.5);
      result.sulfurscalc:=trunc(me/element_data[4].weight);
      for i:=1 to psize do
        p_intensity[i]:=(p_intensity[i]/m)*100;
      j:=-1;
      equal:=false;
      while ((j<7) and not equal) do
        begin
          j:=j+1;
          k:=-1;
          while ((k<4) and not equal) do
            begin
              k:=k+1;
              if (psize=(j+k)) then
                begin
                  equal:=true;
                  temp:=BrCl_table[j,k];
                  i:=0;
                  while((i<psize) and equal and (temp<>nil)) do
                    begin
                      i:=i+1;
                      temp:=temp^.next;
                      if ((temp<>nil) and (p_intensity[i]<=(temp^.data*
```

61

```
                        (1+(percent/100))))  and  (p_intensity[i]>=
                        (temp^.data*(1-(percent/100))))))  then
                       equal:=true
                     else
                       equal:=false;
                 end;
             if ((i=0) and (BrCl_table[j,k]=nil))
               then equal:=false;
             if ((i=psize) and (temp<>nil)) then
              if (temp^.next<>nil) then
               equal:=false;
             if equal then
              begin
                 result.bromines:=k;
                 result.chlorines:=j;
              end
             else
              begin
                 result.bromines:=0;
                 result.chlorines:=0;
              end;
          end;
       end;
     end;
    writeln('calculations finished');
    writeln;
    output_results(result,me);
  end;


( menu1 - Main menu of part 1 )

procedure menu1(var BrCl_table:table; var element_data:element_list;
                var done:boolean);

var
    flag,nothing:boolean;
    choice:integer;
    m,m1,m2,percent:real;
    me:integer;
    p_intensity:intensity_array;
    psize:integer;
    result:result_record;

    begin

    flag:=false;
    nothing:=true;
    repeat
      repeat
        clrscr;
        writeln('                              Molecular Formula Analysis');
        writeln('                                      menu');
        writeln('                              --------------------------');
        writeln;
        writeln('                    1)    Input spectrum data');
        writeln('                    2)    Print results to screen');
        writeln('                    3)    Run routines');
        writeln('                    4)    Print section description to ',
                 'screen');
        writeln('                    5)    return to main menu');
        writeln('                    6)    exit to dos (quit)');
        writeln;
        write('                    Enter choice (must be an integer): ');
        readln(choice);
      until ((choice>0) and (choice<7));
```

```
     case choice of

          1:begin
               clrscr;
               input_data(m,m1,m2,percent,p_intensity,me,psize);
               flag:=true;
               nothing:=false;
            end;
          2:begin
               clrscr;
               if ((not flag) and (not nothing)) then
                  output_results(result,me)
               else
                  begin
                   if nothing then
                     writeln('Please input data first')
                   else
                    begin
                      writeln('Please run option 3 first.  The current ',
                              'input data does not coincide');
                      writeln('with the current output data.');
                     end;
                    writeln('PRESS <RETURN> TO CONTINUE');
                    readln;
                   end;
             end;
          3:begin
               clrscr;
               if (not flag) then
                  begin
                    writeln('Please input new data first');
                    write(' PRESS <RETURN> TO CONTINUE');
                    readln;
                  end
               else
                  begin
                    calculate(m,m1,m2,percent,me,psize,p_intensity,element_data,
                              result,BrCl_table);
                    flag:=false;
                  end;
              end;
          4:main1_description;
          6:done:=true;
         end;
       until (done or (choice=5));
     end;

{ main1 - loads data for part 1, prints decription of part 1, starts up
          menu for part 1 }

procedure main1(var done:boolean);

var
   BrCl_table:table;
   element_data:element_list;

   begin
     clrscr;
     load_data(BrCl_table,element_data);
     main1_description;
     menu1(BrCl_table,element_data,done);
   end;


{********************************************************************}
{ BEGINNING OF PART 2 - SOME PROCEDURES FROM PART 1 ARE CALLED }
```

```pascal
{ resolution - given number num2, and the number of digits after the
               decimal place res, set number equal to num2 with the
               res digits after the decimal place.
               i.e.  num2 = 10.4564323,
                     res = 4
                     yields
                     number = 10.4564          }

procedure resolution(var number:real;num2:real;res:integer);

var
    i:integer;
    val:real;

begin
    i:=0;
    val:=1;
    while(i<res) do
      begin
        val:=val*10;
        i:=i+1;
      end;
    number:=trunc((val*num2) +0.5)/val;
end;


{ set_resolution - sets the resolution variable res with proper prompts }


procedure set_resolution(var res:integer);

begin
 repeat
  clrscr;
  writeln('Choose the number of digits after the decimal point for all');
  writeln('the atomic weights and the ion mass');
  writeln('must be between 0 and 4');
  writeln('0 = 0 digits after decimal point');
  writeln('1 = 1 digit after decimal point');
  writeln('etc....');
  readln(res);
 until res IN[0,1,2,3,4];
 end;


{ print_text - Prints description of Part 2 contained in an external
               data file }

procedure print_text;

  var
    p:page;
    myfile:text;
    j:integer;

  begin
    j:=0;
    open_file(myfile,'combo.txt');
   while (not eof(myfile)) do
    begin
    clrscr;
    while ((j<22) and not (eof(myfile))) do
        begin
          j:=j+1;
          readln(myfile,p[j]);
```

```
            writeln(p[j]);
          end;
      writeln('PRESS <RETURN> TO CONTINUE');
      readln;
      j:=0;
    end;
  end;

{ get_file_name - opens the data file for use in part 2.  The file
                  contains date for all elements.   }

  procedure get_file_name(var myfile:text);

  var filename:line;

    begin
       filename:='combo.dat';
       open_file(myfile,filename);
    end;

{ capital - procedure to capitalize an entire string. useful for later
            use of string comparison }

  procedure capital(var ch:sttype);

  var i:integer;
      temp:sttype;

  begin
    temp:='';
    for i:=1 to length(ch) do
      temp:=temp+upcase(ch[i]);
    ch:=temp;
  end;


{ capitalfirst - capitalizes the first letter of a string and lowercases the
                 rest }

  function capitalfirst(var ch:sttype):sttype;

  var
     i,t2:integer;
     temp:sttype;

  begin
    temp:='';
    temp:=temp+upcase(ch[1]);
    i:=2;
    while (i<=length(ch)) do
    begin
     t2:=ord(ch[i]);
     if ((t2<=ord('Z')) and (t2>=ord('A'))) then
       temp:=temp+chr(t2+(ord('a')-ord('A')))
     else
       temp:=temp+ch[i];
     i:=i+1;
    end;
    capitalfirst:=temp;
  end;

{ print_data - prints individual ion data to screen in a
               pleasing format }


  procedure print_data(ion:datalist;m:dualarray;res:integer);
```

```pascal
var
   i:integer;

begin
    for i:=1 to length(ion.n) do
       write(ion.n[i]);
    for i:=length(ion.n)+1 to 20 do
       write(' ');
    writeln(ion.s:4,ion.w:14:res,m[1]:10,m[2]:10);
end;


{ print_list - prints output of ion list determined in part 2 }

procedure print_list(var ion:ionlist;var max_coeff:double;
                     res,num_entries:integer;number:real);

var
   i,j:integer;

begin
    clrscr;
    i:=1;
    writeln('ion mass = ',number:8:res);
    writeln('name                symbol      weight     max. coeff.  min. ',
            'coeff.');
    writeln('-----------------------------------------------------------------',
            '-------');
    while (i<=num_entries) do
     begin
       j:=1;
       while ((j<20) and (i<=num_entries)) do
         begin
           print_data(ion[i],max_coeff[i],res);
           i:=i+1;
           j:=j+1;
         end;
      writeln('PRESS <RETURN> TO CONTINUE');
      readln;
    end;
end;


{ list_element_data - prints out data currently being used in part 2
                      for specified elements }


procedure list_element_data(var ion:ionlist;var max_coeff:double;
                            res,num_entries:integer);

var
   i:integer;
   ch:sttype;
   found:boolean;

begin
    found:=false;
    clrscr;
    writeln('This routine prints out the data being used for a specified');
    writeln('element.');
   repeat
    writeln('enter q to quit');
    writeln('enter element name or symbol');
    readln(ch);
    capital(ch);
```

```pascal
          if (ch<>'Q') then
          i:=1;
          while ((i <= num_entries) and not found) do
           begin
            if ((ch=ion[i].n) or (ch=ion[i].s))
             then
              begin
               writeln('name                   symbol      weight      max. coeff.  ',
                       'min. coeff.');
               writeln('------------------------------------------------------------',
                       '-------------');
               print_data(ion[i],max_coeff[i],res);
               found:=true;
              end;
            i:=i+1;
           end;
          if (not found and (ch<>'Q')) then
            writeln('element not found in list. try again.');
          until (found or (ch='Q'));
          writeln('PRESS <RETURN> TO CONTINUE');
          readln;

end;

{ change_all_nums_res - changes the resolution of all numbers to
                        a new resolution }

procedure change_all_nums_res(var ion:ionlist;var number:real;
                              var res:integer;elements:elementlist;
                              num_entries:integer);

var
  i,j:integer;

begin
   clrscr;
   writeln('This routine changes the resolution of the ion mass and all ',
           'the');
   writeln('weights of the elements being considered in the calculations.');
   writeln('It is possible for the user to change the ion mass completely,');
   writeln('but this does not affect the maximum element coefficients.');
   writeln('PRESS <RETURN> TO CONTINUE');
   readln;
   set_resolution(res);
   writeln('old ion mass = ',number:8:res);
   writeln('input new ion mass');
   readln(number);
   resolution(number,number,res);
   for j:=1 to num_entries do
     for i:=1 to MAXELE do
       if (ion[j].s=elements[i].s) then
         resolution(ion[j].w,elements[i].w,res);
end;


{ read_data_file - reads in the element data file for use in Part 2 }

procedure read_data_file(var elements:elementlist);

var myfile:text;
    i:integer;

    begin
      open_file(myfile,'combo.dat');
```

```
          i:=1;
          while (i<=MAXELE) do
            begin
              readln(myfile,elements[i].n);
              readln(myfile,elements[i].s);
              readln(myfile,elements[i].w);
              i:=i+1;
            end;
          close(myfile);
        end;


   { test_name - looks up element data given the name of the data and
                  places it in the current ion-list position }

   procedure test_name(str1:sttype;var ion:ionlist;var elements:elementlist;
                    var tn:boolean;res:integer;count:integer);

     var
       i:integer;


     begin
        i:=1;
        tn:=false;
        while ((i<=MAXELE) and (not tn)) do
          begin
            if ((str1=elements[i].n) or (str1=elements[i].s))
              then
                begin
                  tn:=true;
                  ion[count].n:=elements[i].n;
                  ion[count].s:=elements[i].s;
                  resolution(ion[count].w,elements[i].w,res);
                end;
            i:=i+1;
          end;
     end;

{ test_duplicate - checks for duplication of an element being added
                   to the ion-list in Part 2 }

procedure test_duplicate(str1:sttype;var ion:ionlist;count:integer;
                         var found:boolean);

var i:integer;

begin
    i:=1;
    found:=false;
    while ((i<=count) and not (found)) do
      begin
        if((str1=ion[i].n) or (str1=ion[i].s))
          then
            found:=true;
        i:=i+1;
      end;
end;



{ test_weight - compares weight to number, tw set to false if weight
               is larger, true otherwise }

procedure test_weight(weight,number:real;var tw:boolean);
```

68

```
begin
    if (weight>number)
      then
        tw:=false
      else
        tw:=true;
end;

{ add_element - adds elements to the ion-list.  The ion-list is a list
                of elements that are presumed to be in the ion.   }

procedure add_element(var ion:ionlist;var elements:elementlist;count:integer;
                      res:integer;number:real;var flag:boolean);
var
   td,tn,tw:boolean;
   i,j,temp:integer;
   d,ch:char;
   new,temp2:sttype;

   begin
           repeat
                 repeat
                   writeln('input element name or symbol');
                   readln(new);
                   temp2:=capitalfirst(new);
                   test_duplicate(temp2,ion,count-1,td);
                   test_name(temp2,ion,elements,tn,res,count);
                   if (temp2='Q') then
                        begin
                          tn:=true;
                          td:=false;
                        end;
                   if (not tn) then
                        writeln('name incorrect try again');
                   if td then
                        writeln(new,' already added to ion ');
                 until (tn and not td);
                 test_weight(ion[count].w,number,tw);
                 if (temp2='Q') then
                   tw:=true;
                 if (not tw) then
                  begin
                   writeln('element weight is larger than ion weight entered');
                   writeln('try again');
                  end;
           until tw;
             if (temp2<>'Q') then
              begin
                writeln(ion[count].n,' added to list');
                flag:=true;
              end
             else
                begin
                  ion[count].n:=' ';
                  ion[count].s:=' ';
                  ion[count].w:=0.0;
                  flag:=false;
                end;


   end;


{ input_coeff - prompts for maximum and minimum coefficients of an element
                in the ion list. The coefficients are later used as a range
                to search through when calculating a total molecular weight
```

69

```
                of the final ion. }

procedure input_coeff(var name:sttype;var coeff:dualarray);

        begin
          writeln('input MAXIMUM element coefficient for ',name,'.');
          write('(must be an integer) ');
          readln(coeff[1]);
          writeln;
          writeln('input MINIMUM element coefficient for ',name,'.');
          write('(must be an integer) ');
          readln(coeff[2]);
          writeln;
        end;


{ input_coefficients - sets max and min coefficients as discussed in
                        input_coeff }

  procedure input_coefficients(var ion:ionlist;var max_coeff:double;
                               num_entries:integer;number:real);

  var
      incoeff:char;
      count,temp:integer;
      test:boolean;

  begin
      repeat
        writeln('Would you like to input your own bounds on the element ',
                'coefficients');
        writeln('or would like the computer to calculate its own bounds ? ');
        write('(Y)ou or (C)omputer: ');
        readln(incoeff);
        incoeff:=upcase(incoeff);
      until incoeff IN ['Y','C'];
      if (incoeff='Y') then
        begin
          for count:=1 to num_entries do
            begin
              temp:=trunc(number/ion[count].w);
              repeat
               input_coeff(ion[count].n,max_coeff[count]);
               test:=((max_coeff[count][1]>temp) or (max_coeff[count][1]<=0) or
                      (max_coeff[count][2]>temp) or (max_coeff[count][2]<=0) or
                      (max_coeff[count][1]<max_coeff[count][2]));
               if test
                 then
                   begin
                     writeln;
                     writeln('maximum or minimum element coefficient too high or ',
                             'too low');
                     writeln('or the maximum element coefficient is less than ',
                             'the minimum.');
                     writeln('the coefficients must be greater than 0 and less ',
                             'than or equal to');
                     writeln(temp,' and the maximum must be greater than the ',
                             'minimum .  Try again.');
                     writeln;
                   end;
              until (not test);
            end;
        end
      else
       begin
```

```pascal
        writeln('computing maximum and minimum element coefficients');
       for count:=1 to num_entries do
        begin
         max_coeff[count][2]:=1;
         max_coeff[count][1]:=trunc(number/ion[count].w);
        end;
        writeln('PRESS <RETURN> TO CONTINUE');
        readln;
      end;

   end;

{initialize_ion - initializes the ion-list }

procedure initialize_ion(var ion:ionlist);

   var
    i:integer;

   begin
     for i:=1 to MAXSIZE do
       begin
         ion[i].n:=' ';
         ion[i].s:=' ';
         ion[i].w:=0.0;
       end;
    end;


{ ask for all input data for use in part 2 }

   procedure input_test_data(var elements:elementlist;var ion:ionlist;
                             var number:real;var num_entries:integer;
                             var max_coeff:double;var res:integer);

   var
      count:integer;
      flag:boolean;
      answer:char;

      begin
         clrscr;
         flag:=true;
         count:=1;
         write('Please input the ion mass: ');
         readln(number);
         writeln('note:  the number will be rounded off to ',res,' digits ',
                 'after the');
         writeln('decimal point.');
         resolution(number,number,res);
     repeat
       repeat
         repeat
          writeln('enter the number of elements in the ion');
          writeln('must be greater than 0 and less than or equal to ',MAXSIZE);
          readln(num_entries);
         until ((num_entries>0) and (num_entries<=MAXSIZE));
         writeln;
         writeln('input q to quit addition of elements at any time');
         while ((count<=num_entries) and (flag)) do
           begin
             add_element(ion,elements,count,res,number,flag);
             count:=count+1;
           end;
         if (not flag) then
           begin
```

```
                    count:=count-1;
                    num_entries:=count-1;
                    repeat
                      write('Do you wish to reenter the elements ? ');
                      readln(answer);
                      answer:=upcase(answer);
                    until (answer) IN['Y','N'];
                    if (answer='Y') then
                      begin
                        initialize_ion(ion);
                        flag:=true;
                        count:=1;
                      end;
                  end
                else
                    answer:='N';
            until (answer='N');
            if (num_entries=0) then
              begin
                initialize_ion(ion);
                flag:=true;
                count:=1;
                writeln('Sorry.  The number of elements in the ion equals 0');
                writeln('please try again.  This inconvenience is present to ',
                        'prevent');
                writeln('run-time errors');
              end;
          until (num_entries<>0);
              input_coefficients(ion,max_coeff,num_entries,number);
          end;


(get_elements - read in the data for the elements and ask for
                all input data for part 2 }

procedure get_elements(var ion:ionlist;var max_coeff:double;
                        var number:real;var num_entries:integer;
                        var elements:elementlist;var res:integer);

    var
        i:integer;
        c:char;
        nametest:boolean;

        begin
            read_data_file(elements);
            input_test_data(elements,ion,number,num_entries,max_coeff,res);
        end;



{ initializes the max,min coefficients as discussed in input_coeff}

procedure init_coeff(var coeff_count:coefflist;var max_coeff:double;
                     num_entries:integer);

    var
        i:integer;

        begin
            i:=0;
            for i:= 1 to num_entries do
                coeff_count[i]:=MAX_COEFF[i][2];
            for i:=(num_entries+1) to MAXSIZE do
                coeff_count[i]:=0;
        end;
```

```
{ print_elements - prints valid combination data in myfile that is beyond
                    size of the internal combination data array }

procedure print_elements(var coeff_count:coefflist;num_entries:integer;
                         var myfile2:text);

  var
     i:integer;

      begin
         for i:=1 to num_entries do
            writeln(myfile2,coeff_count[i]);
      end;

{ store_coeff - stores valid combination data in combination array }

procedure store_coeff(var combo:combolist;var coeff_count:coefflist;
                      j,num_entries:integer);

var
   i:integer;

begin
    for i:=1 to num_entries do
       combo[j][i]:=coeff_count[i];
end;

{ check_coeff - checks to see if a given coefficient lists results in
                a molecular formula adds up to the ion weight and if so
                stores the coefficients in a data array (or to a file
                depending on the number of valid coefficient lists) for
                later use in output }

procedure check_coeff(var coeff_count:coefflist;number:real;
                      res,num_entries:integer;var ion:ionlist;
                      var found:boolean;var combo:combolist;var j:integer;
                      var key:char; var filename:line);

  var
     i,l : integer;
     sum : real;
     myfile2:text;

      begin
         sum:=0.0;
         i:=0;
         while ((i<num_entries) and (sum<=number)) do
           begin
            i:=i+1;
            sum:=sum+(coeff_count[i]*ion[i].w);
            resolution(sum,sum,res);
           end;
         if ((sum=number) and (i=num_entries))
           then
             begin
               j:=j+1;
               found:=true;
               if (j<=MAXCOMB) then
                begin
                   write(j:4);
                   store_coeff(combo,coeff_count,j,num_entries);
                end
               else
                     if (j>MAXCOMB) then
```

```
                        begin
                          if(j=MAXCOMB+1) then
                            begin
                              writeln;
                              writeln;
                              writeln('more than ',MAXCOMB,' elements found');
                              writeln('opening data file ',filename);
                              assign(myfile2,filename);
                              rewrite(myfile2);
                              writeln('storing the coefficients that have ',
                                      'already been calculated');
                              for l:=1 to MAXCOMB do
                                print_elements(combo[l],num_entries,myfile2);
                              writeln('finished storing');
                              writeln('press <space> and <return> to continue,',
                              ' "Q" and a return to quit.');
                              readln(key);
                              key:=upcase(key);
                              if (key<>'Q') then
                              writeln('running total of combinations found:');
                            end;
                          if (key<>'Q') then
                            begin
                              print_elements(coeff_count,num_entries,myfile2);
                              write(j:4);
                            end;
                        end;

          end;
      end;


  { combinations - guts of Part 2 - an exhaustive search of all possible
                    coefficients ( a combinations)  to find possible
                    molecular formula whose molecular weight add up to
                    the given ion weight.  Various limits(i.e max and min
                    coefficients) are imposed by the user on the element's
                    coefficients to narrow the search. }


  procedure combinations(var ion:ionlist;var max_coeff:double;
                    number:real;res,num_entries:integer;
                    var elements:elementlist;var combo:combolist;
                    var combocount:integer;var filename:line);

   var
    number_iterations,point,j:integer;
    coeff_count : coefflist;
    endfound,found : boolean;
    i:longint;
    key:char;
    myfile2:text;

    begin
      clrscr;
      writeln(' We are about to calculate all possible combinations of the');
      writeln('elements whose atomic weights add up to the ion mass.');
      writeln('In order to quit at anytime, type "Q" followed by a return.');
      writeln('Press any other key, the program will pause until the');
      writeln('return key is pressed.');
      writeln;
      writeln('Note: if the number of combinations exceeds ',MAXCOMB,',');
      writeln('the program will store the data on the disc drive.  Please be ',
              'sure');
      writeln('a disc is in the default drive.  Please enter the name');
      write('you would like this file to have (the default is excess.out): ');
```

```pascal
      readln(filename);
      if (filename='') then
        filename:='excess.out';
        writeln;
        writeln('PRESS <RETURN> NOW TO START EXECUTION');
        readln;
        writeln('running total of combinations found:');
        i:=0;
        point:=num_entries;
        key:=' ';
        combocount:=0;
        found:=false;
        init_coeff(coeff_count,max_coeff,num_entries);
        while ((coeff_count[1]<=max_coeff[1][1]) and (key<>'Q')) do
          begin
            endfound:=false;
            while ((point > 1) and (not endfound) and (coeff_count[1] <=
                max_coeff[1][1])) do
            if (coeff_count[point] > max_coeff[point][1])  then
                begin
                  coeff_count[point]:=max_coeff[point][2];
                  point:=point-1;
                  coeff_count[point]:=coeff_count[point]+1;
                end
            else
                endfound:=true;
            if (coeff_count[1]<=max_coeff[1][1]) then
              begin
                check_coeff(coeff_count,number,res,num_entries,ion,found
                    ,combo,combocount,key,filename);
                i:=i+1;
                point:=num_entries;
                coeff_count[point]:=coeff_count[point]+1;
              end;
            if keypressed then
              begin
                read(key);
                key:=upcase(key);
              end;
          end;
        writeln;
        if not found then
            writeln('no combination found');
        writeln('combination calculation finished.');
        writeln('Press <return> to continue');
        readln;
        if (combocount>MAXCOMB) then
            close(myfile2);
  end;


{ delete_element - deletes an element from the current ion-list }

procedure delete_element(var ion:ionlist;var max_coeff:double;
                          var num_entries:integer);

var
  i:integer;
  ch:sttype;
  found:boolean;

begin
  clrscr;
  writeln('this routine deletes one or more elements from the list of ',
          'possible elements in the');
  writeln('ion, but only one element can be deleted at a time.');
```

```
      repeat
        found:=false;
        writeln('input q to quit deletion');
        writeln('input name or symbol to delete');
        readln(ch);
        writeln(ch);
        ch:=capitalfirst(ch);
        writeln(ch);
        if (ch<>'Q') then
          begin
            i:=1;
            while ((i<=num_entries) and (not found)) do
              begin
                if ((ion[i].n=ch) or (ion[i].s=ch)) then
                  begin
                    writeln(ion[i].n,' deleted');
                    max_coeff[i][1]:=max_coeff[num_entries][1];
                    max_coeff[i][2]:=max_coeff[num_entries][2];
                    ion[i].n:=ion[num_entries].n;
                    ion[i].w:=ion[num_entries].w;
                    ion[i].s:=ion[num_entries].s;
                    max_coeff[num_entries][1]:=0;
                    max_coeff[num_entries][2]:=0;
                    num_entries:=num_entries-1;
                    found:=true;
                  end;
                i:=i+1;
              end;
            if not found then
              writeln('element not found, please try again');
          end;
        if (num_entries=1) then
          begin
            writeln('You cannot delete any more elements.  The number of ',
                    'elements in the list');
            writeln('is currently one.  This inconvenience is present to reduce',
                    ' run-time errors');
            writeln('To delete the last element, please add another element and',
                    ' then come back');
            writeln('and delete the desired element.  Thank you.');
            writeln('PRESS <RETURN> TO CONTINUE');
            readln;
          end;
      until ((ch='Q') or (num_entries=1));
end;

{ change - changes a single specified element's max and min coefficients }

procedure change(var name:sttype; var max_coeff:dualarray;
                 num_entries:integer;weight,number:real;incoeff:char);

var
  temp:integer;
  test:boolean;

begin
  temp:=trunc(number/weight);
  if (incoeff='Y') then
    repeat
      input_coeff(name,max_coeff);
      test:=((max_coeff[1]>temp) or (max_coeff[1]<=0) or
             (max_coeff[2]>temp) or (max_coeff[2]<=0) or
             (max_coeff[1]<max_coeff[2]));
      if test then
        begin
          writeln;
```

```
                writeln('maximum or minimum element coefficient too high or too ',
                        'low');
                writeln('or the maximum element coefficient is less than the ',
                        'minimum.');
                writeln('the coefficients must be greater than 0 and less than or ',
                        'equal to');
                writeln(temp,' and the maximum must be greater than the minimum. ',
                        ' Try again.');
                writeln;
              end;
          until (not test)
        else
          begin
            max_coeff[1]:=temp;
            max_coeff[2]:=1;
          end;
end;

{ change_coeff - changes the max,min coefficients for one or more elements }

procedure change_coeff(var ion:ionlist;var max_coeff:double;
                       num_entries:integer;number:real);

var
    count:integer;
    ch:sttype;
    found,test:boolean;
    temp:integer;

begin
    clrscr;
    writeln('This routine changes the maximum and minimum element ',
            'coefficients for');
    writeln('one or more elements in the ion.  It does not rerun the ',
            'program');
    writeln('with these new element coefficients (see option "4" in menu).');
    repeat
      writeln('input q to quit');
      writeln('input the name or symbol to change coefficient for');
      readln(ch);
      ch:=capitalfirst(ch);
      for count:=1 to num_entries do
        if ((ch=ion[count].n) or (ch=ion[count].s)) then
          begin
            found:=true;
            temp:=trunc(number/ion[count].w);
            repeat
              input_coeff(ion[count].n,max_coeff[count]);
              test:=((max_coeff[count][1]>temp) or (max_coeff[count][1]<=0) or
                     (max_coeff[count][2]>temp) or (max_coeff[count][2]<=0) or
                     (max_coeff[count][1]<max_coeff[count][2]));
              if test then
                begin
                  writeln;
                  writeln('maximum or minimum element coefficient too high ',
                          'or too low');
                  writeln('or the maximum element coefficient is less than ',
                          'the minimum.');
                  writeln('the coefficients must be greater than 0 and less ',
                          'than or equal to');
                  writeln(temp,' and the maximum must be greater than the ',
                          'minimum .  Try again.');
                  writeln;
                end;
            until (not test);
          end;
```

77

```pascal
          if not found then
              writeln('element not found please try again');
      until (ch='Q');
      writeln('PRESS <RETURN> TO CONTINUE');
      readln;
end;

( new_number - changes the current ion molecular weight )

procedure new_number(var number:real;res,num_entries:integer;var max_coeff:
                     double;var ion:ionlist);

var i:integer;

begin
   clrscr;
   writeln('This routine will ask you for a new ion mass.  It does not');
   writeln('rerun the program with this new mass.  To do so, type "4"');
   writeln('when back at the menu screen.  This routine does, however,');
   writeln('change the maximum element coefficients according to machine');
   writeln('calculations.  If you wish to change any one of the ',
           'coefficients');
   writeln('to something different, type "3" when you return to the menu ',
           'screen.');
   writeln;
   writeln('old ion mass = ',number:8:res);
   writeln('input new ion mass');
   readln(number);
   resolution(number,number,res);
   writeln('changing maximum element coefficients');
      for i:=1 to num_entries do
        begin
          max_coeff[i][1]:=trunc(number/ion[i].w);
          max_coeff[i][2]:=1;
        end;
   writeln('PRESS <RETURN> TO CONTINUE');
   readln;
end;


( print_combinations - outputs the possible molecular formula as derived
                        for part 2 )

procedure print_combinations(var ion:ionlist;var combo:combolist;
                             combocount,num_entries,res:integer;number:real;
                             var filename:line);

var
   i,j,k,offset,tab:integer;
   n:char;
   myfile2:text;

begin
   offset:=0;
   i:=1;
   clrscr;
   while ((i+offset)<=combocount) do
    begin
     writeln('                                        elements');
     writeln('                                        --------');
     write('            |   ');
     tab:=trunc((MAXSIZE-num_entries)/2)-1;
     k:=1;
     while (k<=tab) do
      begin
       write('     ');
```

78

```
              k:=k+1;
           end;
        for k:=1 to num_entries do
           write(ion[k].s:4);
        writeln;
        for k:=1 to 79 do
           write('-');
        writeln;
           while (((i+offset)<=combocount) and (i<=18)) do
              begin
                write('        | ');
                if(combocount>MAXCOMB) then
                    begin
                      if ((i+offset)=1) then
                         open_file(myfile2,filename);
                      k:=1;
                      while(k<=tab) do
                       begin
                         write('     ');
                         k:=k+1;
                       end;
                      for k:=1 to num_entries do
                       begin
                          readln(myfile2,j);
                          write(j:4);
                       end;
                      i:=i+1;
                    end
              else
              begin
               j:=1;
               while(j<=tab) do
                begin
                  write('     ');
                  j:=j+1;
                end;
               for j:=1 to num_entries do
                write(combo[i+offset][j]:4);
               i:=i+1;
               writeln;
              end;
             end;
        writeln;
        writeln('PRESS <RETURN> TO PRINT MORE PAGES');
        readln;
        clrscr;
        offset:=offset+(i-1);
        i:=1;
     end;
     if (combocount>MAXCOMB) then
        close(myfile2);
     writeln('no more pages, press <return> to go to menu');
     readln;
end;

{ write_a - prints header for adding element }

procedure write_a;

begin
   writeln('this option adds an element to the list of possible elements');
   writeln('in the ion.');
end;


{ menu2 -  main menu for part 2 }
```

```pascal
procedure menu2(var ion:ionlist;var max_coeff:double;var number:real;
                var num_entries,combocount:integer;var elements:elementlist;
                res:integer; var combo:combolist;var filename:line;var done
                :boolean);

var
  incoeff:char;
  ch:integer;
  flag,add:boolean;

begin
  add:=false;
  repeat
    clrscr;
      repeat
        writeln('                                Possible Ion Combinations Analysis');
        writeln('                                       Menu');
        writeln('                         --------------------------------');
        writeln;
        writeln('                    1) Add element to current ion');
        writeln('                    2) Delete element from current ion');
        writeln('                    3) Change maximum element coefficient on ',
           'list');
        writeln('                    4) Find combinations');
        writeln('                    5) Re-run program with new list of data');
        writeln('                    6) Change ion mass');
        writeln('                    7) Set resolution');
        writeln('                    8) print Table of possible combinations');
        writeln('                    9) list data for Element');
        writeln('                   10) Print list of data being used');
        writeln('                   11) section description');
        writeln('                   12) Exit to Main Menu');
        writeln('                   13) Exit to DOS');
        writeln;
        write('                 choice:  ');
        readln(ch);
      until ((1<=ch) and (ch<=13));
      case ch of
        1:begin
            add:=true;
            clrscr;
            write_a;
            num_entries:=num_entries+1;
            writeln('input q to quit addition of element');
            add_element(ion,elements,num_entries,res,number,flag);
            if (not flag) then
              begin
                writeln('element not added');
                num_entries:=num_entries-1;
              end
            else
              begin
                repeat
                  writeln('Would you like to input a maximum coefficient');
                  readln(incoeff);
                  incoeff:=upcase(incoeff);
                until incoeff IN ['Y','N'];
                change(ion[num_entries].n,max_coeff[num_entries],num_entries
                    ,ion[num_entries].w,number,incoeff);
              end;
            writeln('<return> to continue');
            readln;
          end;
        2:if (num_entries<>1) then
            begin
```

```
                  delete_element(ion,max_coeff,num_entries);
                  add:=true;
               end
             else
               begin
                  writeln('cannot execute.  The number of elements in the ion ',
                          'is one.');
                  writeln('You must add an element first.  This inconvenience ',
                          'is present to prevent');
                  writeln('run-time errors.');
                  writeln('PRESS <RETURN> TO CONTINUE');
                  readln;
               end;
           3:change_coeff(ion,max_coeff,num_entries,number);
           4: begin
                  add:=false;
                  combinations(ion,max_coeff,number,res,num_entries
                      ,elements,combo,combocount,filename);
                  print_combinations(ion,combo,combocount,num_entries,res,
                      number,filename);
              end;
           6:new_number(number,res,num_entries,max_coeff,ion);
           5:begin
                  add:=false;
                  get_elements(ion,max_coeff,number,num_entries
                      ,elements,res);
                  combinations(ion,max_coeff,number,res,num_entries
                      ,elements,combo,combocount,filename);
                  print_combinations(ion,combo,combocount,num_entries,res,
                      number,filename);
              end;
           7: begin
                  add:=true;
                  change_all_nums_res(ion,number,res,elements
                      ,num_entries);
              end;
           8:if (add) then
                  begin
                     writeln('cannot print the table.  You must update it to the ',
                             'current list');
                     writeln('by rerunning the program (5) or Find combinations ',
                             '(4)');
                     writeln('PRESS <RETURN> TO CONTINUE');
                     readln;
                  end
                else
                  print_combinations(ion,combo,combocount,num_entries,res,number
                      ,filename);
           9: list_element_data(ion,max_coeff,res,num_entries);
           10: print_list(Ion,max_coeff,res,num_entries,number);
           11: print_text;
           13: begin
                  writeln('exiting program');
                  done:=true;
               end;
           end;
     until ((done) or (ch=12));
 end;


{ main2 - main procedure for part 2 - first a beginning run-through is
          gone through followed by a menu screen that allows the user
          to change various data and rerun with the new data }

  procedure main2(var done:boolean);
```

81

```
    var
      max_coeff : double;
      num_entries,res,combocount :integer;
      number : real;
      elements:elementlist;
      ion:ionlist;
      combo:combolist;
      filename:line;


    begin
      print_text;
      set_resolution(res);
      get_elements(ion,max_coeff,number,num_entries
            ,elements,res);
      combinations(ion,max_coeff,number,res,num_entries
            ,elements,combo,combocount,filename);
      print_combinations(ion,combo,combocount,num_entries,res,number
            ,filename);
      menu2(ion,max_coeff,number,num_entries,combocount,elements,res,combo
         ,filename,done);
    end;


{ BEGINNING OF PART 3 - THE REFRIGERANT TESTING }

{ add_compounds - adds new compounds to the database of current compounds
                  the procedure asks for the name of the fragment and the
                  m/e ratio of said fragment }

procedure add_compounds;

var
    compound,intensity:string[20];
    myfile:text;
    number:integer;
    ans:char;

begin
  clrscr;
  assign(myfile,'refrig.dat');
  append(myfile);
  repeat
    repeat
      write('Enter name of fragment:  ');
      readln(compound);
      write('Enter ion m/e ratio:  ');
      readln(number);
      write('Enter Intensity:  ');
      readln(intensity);
      writeln('Compound:  ',compound);
      writeln('M/e ratio:  ',number);
      writeln('Intensity:  ',intensity);
      writeln;
      write('Are these correct (Y/N)?  ');
      readln(ans);
      ans:=upcase(ans);
    until(ans='Y');
  writeln(myfile,number,compound:12,intensity:20);
  write('Would you like to input another compound (Y/N)?  ');
  readln(ans);
  ans:=upcase(ans);
  until(ans='N');
  close(myfile);
end;
```

```pascal
{ print_refrig - the title page of part 3 }

procedure print_refrig;
   begin
   end;


{ print_fragment - prints a fragment to stdout - presumably the fragment
                   has been found in the database given the database it's
                   m/e ratio. }

procedure print_fragment(var fragment:fragment_record);

var
   i:integer;

begin
   writeln('FRAGMENT FOUND IN DATABASE');
   writeln('-------------------------');
   writeln('Fragment m/e ratio:   ',fragment.number);
   write('Fragment name:   ');
   for i:=1 to 12 do
     if (fragment.name[i]<>' ') then
         write(fragment.name[i]);
   writeln;
   write('Fragment intensity:   ');
   for i:=1 to 20 do
     if (fragment.intensity[i]<>' ') then
         write(fragment.intensity[i]);
   writeln;
   writeln;
end;


{ search_compounds - searches the refrig database for compounds whose
                     m/e ratio match a given m/e ratio. }

procedure search_compounds;

var
   myfile:text;
   number:integer;
   found:boolean;
   fragment:fragment_record;


begin
   clrscr;
   found:=false;
   open_file(myfile,'refrig.dat');
   reset(myfile);
   write('Please enter m/e ratio to search for:   ');
   readln(number);
   while (not eof(myfile)) do
      begin
          readln(myfile,fragment.number,fragment.name,fragment.intensity);
          if (fragment.number=number) then
            begin
             print_fragment(fragment);
             found:=true;
            end;
      end;
   if (not found) then
       writeln('No matching fragments found in database.');
   close(myfile);
   write('Press return to continue ');
```

```pascal
        readln;
end;


{ refrig - main menu of refrigerant search (Part 3) }

procedure refrig(var done:boolean);

var
    ch,stop:integer;

    begin
        stop:=3;
        print_refrig;
        repeat
         repeat
          clrscr;
          writeln('                         Refrigerant Search');
          writeln('                         ------------------');
          writeln;
          writeln('                         1) Search Fragments');
          writeln('                         2) Add Fragment to database');
          writeln('                         3) Exit to main menu');
          writeln('                         4) Exit to DOS');
          writeln;
          write('                      Choice:');
          readln(ch);
         until((ch>=1) and (ch<=(stop+1)));
        case ch of
            1: search_compounds;
            2: add_compounds;
            4: done:=true;
        end; {case}
       until (done or (ch = stop));
    end;


{ main - main menu of entire program }

procedure main;

var
   done:boolean;
   answer:integer;

begin
    program_description;
    done:=false;
    repeat
     clrscr;
     writeln('                         Fragmentation Analysis');
     writeln('                                Main Menu');
     writeln('                         ----------------------');
     writeln;
     writeln('                      1) Molecular Formula Determination');
     writeln('                      2) Possible Ion Combinations');
     writeln('                      3) Refrigerant Fragments');
     writeln('                      4) Exit to DOS');
     writeln;
     write('            Choice:   ');
     readln(answer);
     case answer of
         1:main1(done);
         2:main2(done);
         3:refrig(done);
         4:done:=true;
```

```
            end;
        until (done);
end;

        begin
            main;
        end.
```

Appendix 3

Vortex-cooled chromatographic syringe

ABSTRACT

This summary describes a simple technique which allows syringe injection to be used with low boiling point liquids such as the chlorofluorocarbons. The method involves the continuous chilling of a standard chromatographic syringe using the cold air stream of a Ranque-Hilsch vortex tube. The cold air is passed through an insulated, flexible tube to a Pyrex envelope which surrounds the syringe. By appropriately "tuning" the vortex tube, syringe temperatures as low as -40 °C can be obtained in five minutes, and maintained at this temperature indefinitely. This technique has been successfully applied to the analysis of a series of new ozone-benign refrigerants having normal boiling points as low as 20 °C. The chilled syringe method has resulted in injection repeatability far better than that obtained using standard syringes at room temperature because of the substantial drop in sample loss due to evaporation.

BACKGROUND

The application of gas chromatography to quantitative analysis requires careful attention to the many sources of error. One of the major contributors to the overall error is the imperfect repeatability of the size of sample injected [1,2], especially when a syringe is used with liquid samples. While internal standardization will remove most of the error associated with imperfect sample size repeatability (of course, selective evaporation always remains a pitfall), optimal performance is obtained only when the sample is maintained constant within experimental error. When external standardization is employed (as is often necessary), the precise repeatability of sample size becomes an imperative [1,2].

The use of a 10 µL chromatographic syringe to deliver a few microliters of a liquid sample can introduce an error of up to 5 percent (in volume, and ultimately in peak area). Not only must one consider the uncertainty of the syringe graduations, evaporation (selective or uniform) and needle

volume contribute to the error. When extremely volatile liquids (such as refrigerants) are being studied, the situation becomes much worse. For the purpose of this discussion, we shall consider a liquid to be volatile if the vapor pressure is above 0.01 MPa (1 psia) at ambient temperature, or if its normal boiling point is below 70 °C. It is not unusual for 99 percent of the sample to evaporate in the 2 or 3 seconds it takes an operator to sample and inject a volatile liquid, even if the syringe is precooled in a refrigerator. The use of a gas sampling valve, while attractive for several reasons, is often impossible because of limitations on the quantity of sample.

TECHNICAL DESCRIPTION

Components

The apparatus shown in figure 3 has been designed and constructed to address the problem of evaporation of volatile liquids from chromatographic syringes. Although the approach here is specifically for volatile liquids, it can be employed to reduce syringe evaporation in any application. A standard 10 µL syringe is fitted with a pyrex envelope (2 cm o.d., 0.16 cm wall thickness). The envelope, which has an inlet and an outlet tubulation, is sealed to the syringe using short lengths of Tygon tubing. The envelope has two strips of Velcro on its sides to allow easy handling and stowage. The lower (inlet) tubulation is fitted with a length of Tygon tubing which has been covered with an insulating jacket. This jacket is made up of three layers of glass fiber insulation each of which has a slightly larger inside diameter. This tube so insulated is connected to the cold exit of a Ranque-Hilsch vortex tube [3-8], which is itself insulated with ordinary pipe insulation. The vortex tube is a unique and valuable heating and cooling device which has recently found great usefulness in chemical laboratories [9-13]. The operating principles and control parameters of the vortex tube have been discussed in detail elsewhere [5-7], and the reader is referred to these sources for details. The tube used in the present application is a commercial device having a flow capacity of 0.07 m^3/min. The

line which carries the air from the cold air exit has an inside diameter which is slightly larger than the inside diameter of the cold air exit itself. This arrangement introduces no appreciable back pressure to the cold air exit, which is a very important consideration for the proper functioning of the vortex tube. The hot exit of the vortex tube is directed into a baffle muffler which vents above the ceiling of the laboratory. This is done strictly as a noise control measure, since in normal operation of the cooled syringe most of the air flow (60 - 80 percent) will be vented from the hot end of the tube. Other noise control measures include surrounding the vortex tube with a lead-layered sound barrier sheet, which eliminates all but the highest frequencies. The muffler and sound control sheet reduce the noise level to a comfortable 65 d.

Operation

Ordinary laboratory compressed air is applied to the inlet of the vortex tube (0.76 MPa, 110 psi) after it is passed through a 10 µm particulate filter and two oil separators (coalescence filters). The coalescence filters are necessary to remove condensed water and oil droplets from the air stream. The service air should be as dry as possible, usually requiring that the laboratory compressor installation be equipped with a dew point dryer. After approximately five minutes of vortex tube operation, the syringe can be equilibrated at a temperature as low as -40 °C. The Velcro strips on the sides of the envelope allow comfortable handling, and prevent heat from the operator's fingers from effecting the temperature of the syringe. With the syringe continuously chilled, one can easily sample and inject volatile liquids such as refrigerants with minimal concern about sample evaporation.

EXAMPLE OF APPLICATION

The vortex cooled syringe injection technique was used in the trace water analysis of three new ozone-benign refrigerants. The refrigerants studied were 1,1,2-trifluoro-1,2-dichloroethane (R123a), 1,1,1-trifluoro-1,1-dichloroethane (R123), and 1-fluoro-1, 1-dichloroethane (R141b). The analyses were done using Porapak-QS (80/100 mesh, in a 2 m packed column, 0.32 cm o.d.) at a column temperature of 140 °C, with helium carrier gas and thermal conductivity detection. External standardization was done (using solutions prepared from dry acetonitrile) due to the relatively small amount of sample available for analysis and the difficulties involved in internally standardizing such a volatile liquid.

Using the cooled syringe allows far better reproducibility, as can be seen from the example given in table 1. The data cited are for five separate analyses of R141b (with a normal boiling point of 32 °C). The volume listed for the cooled syringe was an average of readings made from the syringe barrel; for the uncooled syringe, volume readings were not possible due to rapid evaporation. The water peak area listed for the uncooled syringe is not simply imprecise but obviously inaccurate, being nearly 40 percent lower than in the case of the cooled syringe.

APPLICATION

The apparatus and techniques which have been discussed in this appendix have application in all analyses of volatile liquids, since the method dramatically increases the reproducibility of the injection process without introducing additional time, work or inconvenience for the operator. The method and apparatus will also be applicable to analyses involving relatively nonvolatile liquids as well, as it decreases selective evaporation of solvents, thus increasing the precision of both internal and external standards calibrations.

Table 1. Example data from the analysis of R141b

Cooled Syringe:

- Sample volume: 5.78 ± 0.05 μL

  (coefficient of variation: 0.93%)

- Water peak area: 944.32 ± 3.35 counts

  (coefficient of variation: 0.35%)

Uncooled Syringe:

- Sample volume: not measurable

- Water peak area: 586.71 ± 217.41 counts

  (coefficient of variation: 37%)

REFERENCES

[1] Grob, R.L., ed., <u>Modern Practice of Gas Chromatography</u>, 2nd ed. John Wiley & Sons, New York, 1985.

[2] Katz, E., ed. <u>Quantitative Analysis Using Chromatographic Techniques</u>, John Wiley & Sons, Chichester, 1987.

[3] Ranque, M.G., J. Phys. Rad., 4, 112, 1933.

[4] Hilsch, R., Rev. Sci. Inst., 18(2), 108, 1943.

[5] Blaber, M.P., J. Sci Instr., 27(6), 1968, 1950.

[6] Blatt, T.A., Truch, R.B., ASME Paper 62-WA-200, 1963.

[7] Jordan, R.C., Priester, G.B., <u>Refrigeration and Air Conditioning</u>, 2nd. ed., Prentice Hall, Englewood Cliffs, NJ, 1948.

[8] Abbott, M.M., Van Ness, H.C., <u>Thermodynamics (Schaum's Outline Series)</u>, McGraw Hill Book Co., New York, 1972.

[9] Bruno, T.J., Anal. Chem. 58, 1596, 1986.

[10] Bruno, T.J., Liq. Chromatogr. HPLC Mag. 4(2), 134, 1985.

[11] Bruno, T.J., J. Chem. Educ., 64(11), 987, 1987.

[12] Dunham, L., Hewlett-Packard Corp., private communication, 1988.

[13] Johnston, M., University of Colorado, Boulder, private communication, 1987.

Appendix 4

Refrigerant code numbers for ethane- and ethylene-based compounds

The digits in the ANSI/ASHRAE coding system [1] are determined from the right—hand side of the number.

1. The first digit on the right is the number of fluorine atoms on the molecule. When there are no fluorines on the molecule, such as in methylene chloride, a zero is assigned to this position.

2. The second digit from the right is one more than the number of hydrogen atoms on the molecule.

3. The third number from the right is one less than the number of carbon atoms in the compound. When this digit is zero, as in the case of methane—based fluids, it is omitted.

4. The number of chlorines is found by substracting the sum of the number of fluorine and hydrogen atoms from the total of the number of atoms that can be connected to the carbon atoms.

5. For cyclic compounds, an upper case "C" is placed before the identifying number.

6. In those instances where bromine is present in place of several or all of the chlorine atoms, the same rules apply except that the letter B after the nurefrigerant shows the presence of bromine. The number to the right of the "B" indicates the number of bromine atoms present. When isomers of brominated compounds exist, the position of the bromine atoms are indicated by the Greek letters α and β. The Greek letters indicate the carbon atoms of the backbone chain, starting from the

end carbon having the largest sum of atomic weights bonded to it. If more than one bromine is bonded to the same carbon, the Greek letter identifying that carbon is repeated for each bromine atom.

7. In the case of isomers, each isomer carries the same number. The different isomers are then indicated by appending lower case a, b, c, etc., to the extreme right–hand side of the code. The most symmetrical isomer does not have a letter appended. The letters are appended to the isomers as they become more and more unsymmetrical. The symmetry is determined by adding the atomic weights of the atoms bonded to each carbon, and subtracting one sum from another.

8. Unsaturated compounds are indicated by the fourth number from the right, which indicates the number of double bonds.

Reference

[1] Number Designation and Safety Classification of Refrigerants, ASHRAE standard, ANSI/ASHRAE 34–1989, American Society of Heating, Refrigerating and Air–Conditioning Engineers, Atlanta, GA, 1989.

BL-114A (5-90)	U.S. DEPARTMENT OF COMMERCE NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY ## BIBLIOGRAPHIC DATA SHEET	1. PUBLICATION OR REPORT NUMBER NIST/TN-1340
		2. PERFORMING ORGANIZATION REPORT NUMBER
		3. PUBLICATION DATE August 1990

4. TITLE AND SUBTITLE

Strategy for Chemical Analysis of Alternative Refrigerants

5. AUTHOR(S)

Thomas J. Bruno

6. PERFORMING ORGANIZATION (IF JOINT OR OTHER THAN NIST, SEE INSTRUCTIONS) **U.S. DEPARTMENT OF COMMERCE** **NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY** **BOULDER, COLORADO 80303-3328**	7. CONTRACT/GRANT NUMBER
	8. TYPE OF REPORT AND PERIOD COVERED

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (STREET, CITY, STATE, ZIP)

10. SUPPLEMENTARY NOTES

11. ABSTRACT (A 200-WORD OR LESS FACTUAL SUMMARY OF MOST SIGNIFICANT INFORMATION. IF DOCUMENT INCLUDES A SIGNIFICANT BIBLIOGRAPHY OR LITERATURE SURVEY, MENTION IT HERE.)

The chemical purity of samples of alternative refrigerants which are used in the determination of thermophysical properties is of paramount importance. Many properties, such as vapor-liquid equilibria and transport properties such as thermal conductivity, require that measurements be performed on as pure a material as possible. Some common impurities such as water can have an extremely deleterious effect on a measurement, while the effects of other impurities are more subtle. Alternative refrigerants which are not available at purity levels that are normally considered "research grade" must be characterized in order for the data to be properly interpreted, and for the reported results to be used appropriately. In addition, it is necessary to assess the reactivity of alternative refrigerants in the presence of many of the common construction materials used in experimental apparatus. Many of the new fluids being considered as replacement refrigerants are known to be very aggressive towards metals and elastomers, especially at elevated temperatures and pressures. The potential for reaction must be considered before measurements are made in an expensive apparatus. During the course of studies on the thermophysical properties of alternative refrigerants, the National Institute of Standards and Technology has evolved an analytical protocol which is applied to each system. In this paper, the various analytical methods will be discussed, and the data of general usefulness presented. In addition, some of the novel instrumental approaches employed at NIST will be described.

12. KEY WORDS (6 TO 12 ENTRIES; ALPHABETICAL ORDER; CAPITALIZE ONLY PROPER NAMES; AND SEPARATE KEY WORDS BY SEMICOLONS)

alternative refrigerants; chemical analysis; gas chromatography; infrared spectrophotometry; mass spectrometry; water analysis

13. AVAILABILITY	14. NUMBER OF PRINTED PAGES
X UNLIMITED	104
FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NATIONAL TECHNICAL INFORMATION SERVICE (NTIS).	
X ORDER FROM SUPERINTENDENT OF DOCUMENTS, U.S. GOVERNMENT PRINTING OFFICE, WASHINGTON, DC 20402.	15. PRICE
ORDER FROM NATIONAL TECHNICAL INFORMATION SERVICE (NTIS), SPRINGFIELD, VA 22161.	

ELECTRONIC FORM

☆U.S. GOVERNMENT PRINTING OFFICE: 1990-573-555/45016

NIST Technical Publications

Periodical

Journal of Research of the National Institute of Standards and Technology—Reports NIST research and development in those disciplines of the physical and engineering sciences in which the Institute is active. These include physics, chemistry, engineering, mathematics, and computer sciences. Papers cover a broad range of subjects, with major emphasis on measurement methodology and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Institute's technical and scientific programs. Issued six times a year.

Nonperiodicals

Monographs—Major contributions to the technical literature on various subjects related to the Institute's scientific and technical activities.

Handbooks—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications—Include proceedings of conferences sponsored by NIST, NIST annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

Applied Mathematics Series—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a worldwide program coordinated by NIST under the authority of the National Standard Data Act (Public Law 90-396). NOTE: The Journal of Physical and Chemical Reference Data (JPCRD) is published quarterly for NIST by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements are available from ACS, 1155 Sixteenth St., NW., Washington, DC 20056.

Building Science Series—Disseminates technical information developed at the Institute on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NIST under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The standards establish nationally recognized requirements for products, and provide all concerned interests with a basis for common understanding of the characteristics of the products. NIST administers this program as a supplement to the activities of the private sector standardizing organizations.

Consumer Information Series—Practical information, based on NIST research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.

Order the above NIST publications from: Superintendent of Documents, Government Printing Office, Washington, DC 20402.

Order the following NIST publications—FIPS and NISTIRs—from the National Technical Information Service, Springfield, VA 22161.

Federal Information Processing Standards Publications (FIPS PUB)—Publications in this series collectively constitute the Federal Information Processing Standards Register. The Register serves as the official source of information in the Federal Government regarding standards issued by NIST pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

NIST Interagency Reports (NISTIR)—A special series of interim or final reports on work performed by NIST for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Service, Springfield, VA 22161, in paper copy or microfiche form.