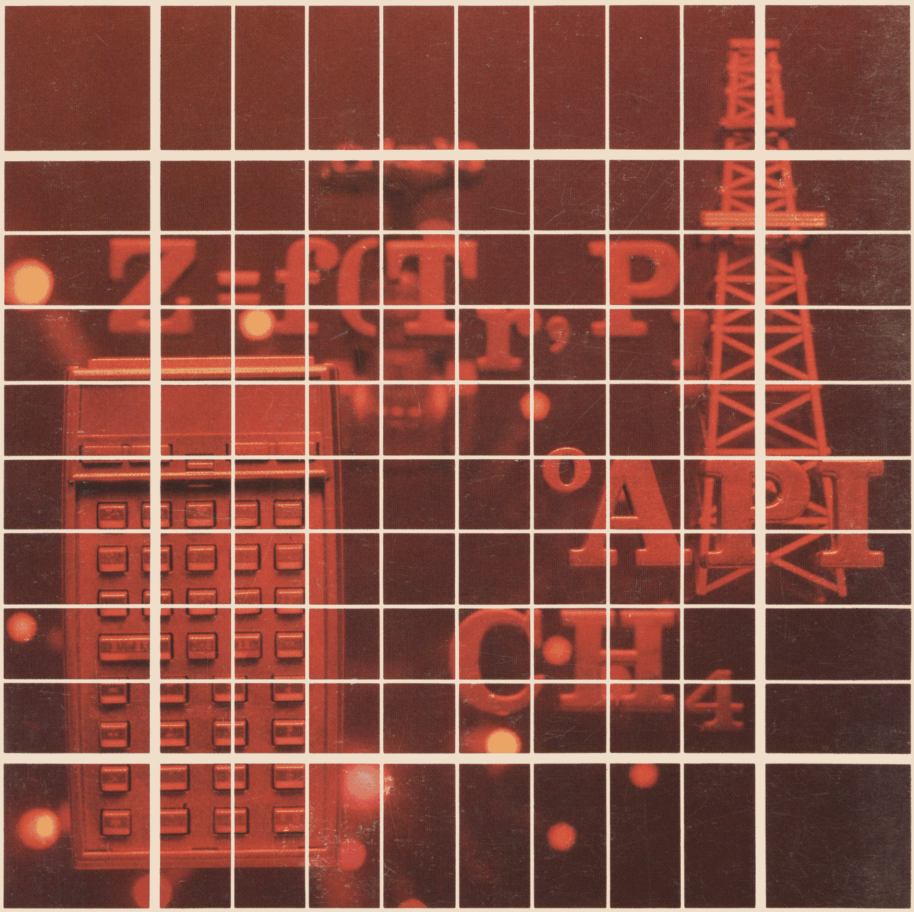
HEWLETT-PACKARD

HP-41C

PETROLEUM FLUIDS PAC



NOTICE

Hewlett-Packard Company makes no express or implied warranty with regard to the keystroke procedures and program material offered or their merchantability or their fitness for any particular purpose. The keystroke procedures and program material are made available solely on an "as is" basis, and the entire risk as to their quality and performance is with the user. Should the keystroke procedures or program material prove defective, the user (and not Hewlett-Packard Company nor any other party) shall bear the entire cost of all necessary correction and all incidental or consequential damages. Hewlett-Packard Company shall not be liable for any incidental or consequential damages in connection with or arising out of the furnishing, use, or performance of the keystroke procedures or program material.

INTRODUCTION

The HP-41 Petroleum Fluids Pac is designed to aid in the rapid estimation of petroleum reservoir fluid properties. The Pac may be divided into three sections: natural gases, oils, and reservoir brines.

All three sections contain programs which estimate the isothermal coefficient of compressibility, formation volume factor, and viscosity.

The programs in the natural gas section also allow the computation of pseudocritical temperature and pressure, gas gravity, heating values, and specific heats.

The oil section programs also determine gas-oil ratio, bubble point pressure, and two-phase formation volume factor. The section on water also includes a program to estimate gas-water ratio.

Two other programs in the Pac estimate rock compressibility and total isothermal coefficient of compressibility for a gas-, oil-, and water-saturated formation.

This Pac has many useful features that provide an added dimension of friendliness and utility to calculator software. These capabilities are illustrated in various sections of the manual. The full power of the Pac will be available to you if you take the time to try Example 1 of each program, and to read *The Petroleum Engineering Unit Management System.*

Each program in this Pac is represented by one program in the Application Module and a section in this manual. Each section includes the following:

- A description of each program.
- The equations on which the program is based.
- Instructions for calling the calculation portion of the program as a subroutine.
- The range of inputs over which the calculated results are valid.
- References for further information on the theory behind the calculations performed or the correlations used.
- A set of instructions for using the program.
- Example problems, each of which includes a list of the keystrokes required for its solution.

Before plugging in your Application Module, **turn your calculator off**, and be sure you understand the section *Inserting and Removing Application Modules.* Before using a particular program, take a few minutes to read *Format of User Instructions,* and *A Word About Program Usage.*

You should first familiarize yourself with a program by running it once or twice while following the complete User Instructions in the manual. Thereafter, the program's prompting should provide the necessary instructions, including which variables are to be input, which keys are to be pressed, and which values will be output.

We hope this Pac will assist you in the solution of numerous problems in your discipline. As you become familiar with the Pac, please feel free to send us your comments and suggestions about this Pac or about other solutions programs you would like to see. Send your comments and suggestions to:

Hewlett-Packard
Corvallis Division Customer Support
1000 N.E. Circle Blvd.
Corvallis, OR 97330, U.S.A.

If you have technical problems with this Pac or with the HP-41, consult your owner's handbook.

Note: Application modules are designed to be used in both HP-41C and HP-41CV model calculators. The HP-41C and HP-41CV differ only in their initial Continuous Memory capacities. The term "HP-41C" is used throughout the rest of this manual, unless otherwise specified, to refer to both calculators.

Acknowledgements:

Hewlett-Packard wishes to thank Mr. D. N. Meehan of Champlin Petroleum Company and Dr. H. J. Ramey of the Department of Petroleum Engineering at Stanford University for their assistance in the definition, implementation, and review of the material in this Pac.

TABLE OF CONTENTS

INSERTING AND REMOVING APPLICATION MODULES

Before you insert an Application Module for the first time, familiarize yourself with the following information.

Up to four Application Modules can be plugged into the ports on the HP-41. While plugged in, the names of all programs contained in the Module can be displayed by pressing [f] CATALOG 2.

CAUTION

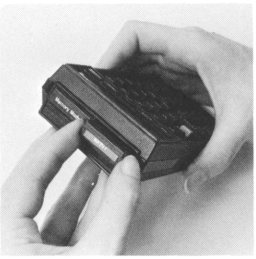
Always turn the HP-41 off before inserting or removing any plug-in extension or accessories. Failure to turn the HP-41 off could damage both the calculator and the accessory.

To insert Application Modules:

1. Turn the HP-41 off! Failure to turn the calculator off could damage both the Module and the calculator.

2. Remove the port covers. Remember to save the port covers; they should be inserted into the empty ports when no extensions are inserted.

3. Insert the Application Module with the label facing downward as shown, into any port **after** the last Memory Module. For example, if you have a Memory Module inserted in port 1, you can insert an Application Module in any of ports 2, 3, or 4. (The port numbers are shown on the back of the calculator.) **Never insert an Application Module into a lower numbered port than a Memory Module.**



4. If you have additional Application Modules to insert, plug them into any port after the last Memory Module. Be sure to place port covers over unused ports.
5. Turn the calculator on and follow the instructions given in this book for the desired application functions.

To remove Application Modules:

1. Turn the HP-41 off! Failure to do so could damage both the calculator and the Module.
2. Grasp the desired Module handle and pull it out as shown.

3. Place a port cap into the empty ports.

Mixing Memory Modules and Application Modules

Any optional accessories (such as the HP 82104A Card Reader, or the HP 82153A Wand) should be treated in the same manner as Application Modules. That is, they can be plugged into any port after the last Memory Module. Also, the HP-41 should be turned off prior to insertion or removal of these extensions.

The HP-41 allows you to leave gaps in the port sequence when mixing Memory and Application Modules. For example, you can plug a Memory Module into port 1 and an Application Module into port 4, leaving ports 2 and 3 empty.

FORMAT OF USER INSTRUCTIONS

The User Instruction Form—which accompanies each program—is your guide to operating the programs in this Pac.

The form is composed of five labeled columns. Reading from left to right, the first column, labeled STEP, gives the instruction step number.

The INSTRUCTIONS column gives instructions and comments concerning the operations to be performed.

The INPUT column specifies the input data or the appropriate ALPHA response to a prompted question. Data input keys consist of 0 to 9 and the decimal point (the numeric keys), [EEX] (enter exponent), and [CHS] (change sign).

The FUNCTION column specifies the keys to be pressed after keying in the corresponding input data.

The DISPLAY column specifies prompts, intermediate and final answers, and their units, where applicable.

Above the DISPLAY column is a box which specifies the minimum number of data storage registers necessary to execute the program. Refer to the Owner's Handbook for information on how the [SIZE] function affects storage register configuration.

The following illustrates the User Instruction Form for the *Gas Isothermal Compressibility* program.

				SIZE: 018
STEP	**INSTRUCTIONS**	**INPUT**	**FUNCTION**	**DISPLAY**
1	Initialize program.		[XEQ] CG	Tc=?
2	Key in pseudocritical temperature.	Tc	[R/S]	Pc=?
3	Key in pseudocritical pressure.	Pc	[R/S]	T=?
4	Key in temperature.	T	[R/S]	P=?
5	Key in pressure and calculate CG.	P	[R/S] [R/S]*	CG=† P=?
6	For a new pressure, go to step 5. For a new case, go to step 1. Key in only values which change. For values which do not change, ignore the prompts and press [R/S].			
	* Press [R/S] if you are not using a printer. †Press [ALPHA] to see the units if you are not using a printer.			

A WORD ABOUT PROGRAM USAGE

Using This Manual

Here are several tips that will help you understand the various parts of the manual for the Petroleum Fluids Pac.

1. Unless specified differently in the examples, all examples assume flags 09 and 10 are clear, the display format is [FIX] 4, and the display mode is flag 28 set and flag 29 clear. (The importance of flags 09 and 10 is discussed in the section on *The Petroleum Engineering Unit Management System.*)

2. Each variable is given a name that corresponds to the petroleum industry symbol for that variable. All the variables, their names, and their default English and SI units are listed for easy reference in Appendix C. For oil property variables, the name refers to the property above the bubble point. Properties at or below the bubble point have a "BP" or a "b", respectively, on the end of this name. The name of each program is the same as the variable being calculated by the program. Oil property programs have the name of the variable above the bubble point.

3. In every section of the manual that describes a program, the first example shows how to use the program. Subsequent examples illustrate the use of Pac subroutines for users wanting to create their own programs. If you do not plan to write your own programs, you should ignore these examples. They will only confuse you with unnecessary details about the internal structure of the Pac. If you do plan to write your own programs using the Pac subroutines, refer to *General Purpose, Input, and I/O Subroutines for Programmers*.

4. If you run a program, and it halts with the following display:

   ***SIZE*>=nnn**

   it means you need at least nnn data registers to run the program. Set the size needed by pressing [XEQ] [ALPHA] SIZE [ALPHA] nnn. Then press [R/S] to continue running the program.

5. The "Range of Validity" section that accompanies each program description is a guide to the range of inputs that will yield an output with a reasonable error. In general, this is the range of values typically encountered in petroleum engineering problems. No error checking is done on these validity ranges. You can use the programs outside the range of validity, but the error may not be acceptable. In some cases, the correlation used will not model real fluid behavior at all outside the range of validity. In those cases where the validity range must not be exceeded, the program will halt with an appropriate error message.

6. Unless otherwise noted, the "Equations" section of each program assumes that all the variables are in Pac English default units.
7. Normally, separator pressures are given in gauge pressure units. Consistent with all pressures used in the Pac, the separator pressures you key in must be in absolute pressure units.
8. Five of the programs estimate brine properties as a function of salinity. For those programs, the salt content is assumed to be 100% sodium chloride.
9. The programs are intended to provide a means to make rapid **estimates** of petroleum fluid properties. The programs are based primarily on empirical correlations. If you have actual fluid property data from an analysis, you should use that instead.

Catalog

When an Application Module is plugged into a port of the HP-41, the contents of the Module can be reviewed by pressing [■] [CATALOG] 2 (the Extension Catalog). Executing the [CATALOG] function lists the name of each program or function in the Module, as well as functions of any other extensions which might be plugged in.

ALPHA and USER Mode Notation

This manual uses a special notation to signify ALPHA mode. Whenever a statement on the User Instruction Form is printed in gold, the [ALPHA] key must be pressed before the statement can be keyed in. After the statement is input, press [ALPHA] again to return the calculator to its normal operating mode, or to begin program execution. For example, [XEQ] Z means press [XEQ] [ALPHA] Z [ALPHA].

When the calculator is in USER mode, this manual will use the symbols [A]—[J] and [■] [A]—[■] [E] to refer to the reassigned keys in the top two rows. These key designations will appear on the User Instruction Form and in the keystroke solutions to sample problems.

Optional Printer

When an optional printer is plugged into the HP-41 along with the Petroleum Fluids Application Module, inputs and results will be printed automatically.

Downloading Module Programs

If you wish to trace execution, to modify, or to record on magnetic cards a program in this Application Module, it must first be copied into the HP-41's program memory. For information concerning the HP-41's [COPY] function, see the Owner's Handbook. It is **not** necessary to copy a

program in order to run it. Some programs in this Pac are written in machine language and may not be copied.

Note that since all the programs in the Petroleum Fluids Pac call subroutines in the Pac, **you will still need the Petroleum Fluids Application Module plugged into a port in order to run a copied program.**

Use of Labels

You should generally avoid writing programs into the calculator memory that use program labels identical to those in your Application Module. In case of a label conflict, the label within program memory has priority over the label within the Application Module.

This feature can be used to substitute your preferred fluid property correlations for those used in the Pac. See item 6 in the *Block Structure* section of *General Purpose, Input, and I/O Subroutines for Programmers.*

Label Conflicts With Other Application Pacs

Five labels used in the Petroleum Fluids Pac have the same name as those used in other Pacs. If you have this Pac and another Pac plugged into your HP-41 at the same time, you should make sure that the Pac whose programs you want to use is in the **lowest-numbered port** to avoid conflicting use of these labels.

Label	Pac
CG	Aviation
COMP	Surveying
RS	Circuit Analysis
OUT	Real Estate
P	Games, Navigation, Standard

Assigning Program Names

Key assignments to keys [A]—[J] and [■][A]—[■][E] take priority over the automatic assignments of local labels in the Application Module. Be sure to clear previously assigned functions before executing a Module program.

THE PETROLEUM ENGINEERING UNIT MANAGEMENT SYSTEM

In many applications, the difficulty of computation is secondary to the difficulties of unit conversions and dimensional homogeneity. The programs in the HP-41 Petroleum Fluids Pac were written to solve both the computational and the dimensional aspects of your problems.

Responding to Input Prompts

Suppose a program prompts you for an input of pressure, as follows:

P=?

You would respond by keying in a number and, optionally, the units associated with that number. The desired units would be input by pressing the [ALPHA] key and spelling the Pac's abbreviation for the units. Then press [R/S] to continue running the program.

For a pressure of 103 atmospheres, you would respond with the following:

103 [ALPHA] ATM [R/S]

Inputs With the Printer

If the optional printer is plugged into the HP-41, the inputs will be printed. For the above example, the input would be printed as follows:

```
P=103.0000 ATM
```

Responding to Output Prompts

The units of outputs may also be specified. The program will beep, stop, and prompt you for units where required. For a pending output of gas formation volume factor (BG), the following would be displayed:

BG, FT3/SCF?

This prompt means that the next output will be BG in cubic feet per standard cubic foot. If FT3/SCF is acceptable, simply press [R/S]. If you prefer some other unit, key it in and press [R/S].

For a formation volume factor in BBL/SCF, you would respond with the following:

BBL/SCF [R/S]

The program halted in ALPHA mode, so you do not need to press the [ALPHA] key before you key in the units.

Outputs With and Without the Printer

If the optional printer is plugged into the HP-41, the outputs will be printed. For the above example, the output would be printed as follows:

```
BG=0.0020 BBL/SCF
```

If the printer is not plugged in, the program will stop to show you the output, like this:

BG=0.0020

When the program halts, the units will be in the ALPHA register. To see the units, press [ALPHA].

BBL/SCF

Then press [R/S] to continue running the program.

Changing Output Units Using Flag 10

The Pac uses flag 10 to control whether or not the program stops to allow you to change the output units. If you want the program to stop and prompt you for output units as illustrated above, set flag 10 ([SF] 10).

If you do not want the program to stop on output, but instead to select units automatically for you, clear flag 10 ([CF] 10). The units selected automatically will be either English or SI units, as will be discussed shortly.

The status of flag 10 (set or clear) is remembered by the HP-41 when it is turned off, and is never altered by any program in the Pac. This means you only have to set or clear the flag once. You will probably set or clear it periodically while using the programs in the Pac, depending on whether or not you want to specify units other than those selected automatically. Example 1 of the *Oil Isothermal Compressibility* program shows how to use flag 10 in this manner. Example 1 of the *Gas Isothermal Compressibility* and *Water Viscosity* programs also illustrate the use of flag 10.

Default Units

Every variable used in the Pac has either English or SI units assigned to it by default; that is, automatically assigned. These default units are shown by category in Table 1. When you are responding to an input prompt, the default units for that input are in the ALPHA register. If you are not sure what unit has been assigned to the input variable, press [ALPHA] to check. If the default unit is acceptable, press [R/S]. If you want a different unit, key it in as shown before, and then press [R/S]. (If

the display is blank when you press [ALPHA], it means the variable is dimensionless, and no units are required.)

For the example at the beginning of the section, when you pressed [ALPHA] in response to the ***P=?*** prompt, you would have seen ***PSI***, the default unit for pressure.

As you have seen, when you respond to an output prompt, the units are displayed as part of the prompt. If flag 10 is clear, the default units will be automatically selected for you, since the program will not halt to allow you to change them.

Whenever either an input or output prompt appears, the number in X will **always** correspond to the units shown in the ALPHA register. This is illustrated in Example 1 of the *Water Viscosity* program.

In response to an input or output prompt, you can use any combination of units from Table 2 that are dimensionally consistent with the default units for that variable.

Example 1 of the *Z Factor* program illustrates in detail the use of the Unit Management System for responding to input prompts with and without default units.

Table 1: Pac Default Units By Category

	English Default Units		SI Default Units	
Quantity	**Abbreviation**	**Meaning**	**Abbreviation**	**Meaning**
Pressure	PSI	Pound per square inch absolute	KPA	Kilopascal
Critical and pseudocritical temperature	R	Degree Rankine	K	Kelvin
All other temperatures	F	Degree Fahrenheit	C	Degree Celsius
Amount of gas	SCF	Standard cubic foot	SCM	Standard cubic meter
Liquid volume	BBL	Barrel of petroleum	M3	Cubic meter of petroleum
Oil gravity	API	Degree API	KG/M3	Kilogram per cubic meter
Viscosity	CP	Centipoise	PA*S	Pascal-second
Energy	BTU	British thermal unit	KJ	Kilojoule

Using the Same Values for Input Variables

There is a common set of input variables used by all the programs in the Pac. **Each variable has its own unique storage location, and this location is always reserved for that variable.** This is extremely useful, because it means that if several programs use the same values for their input variables, those values only need to be entered once. All other times they are requested by an input prompt, you can just press [R/S], and the previously stored value will be retained. This is illustrated in Example 1 of the *Pseudocritical Temperature and Pressure from Gas Gravity* program, and in Example 3 of the *Oil Formation Volume Factor* program.

Whenever you see an input prompt, you can see what the current value of that variable is by pressing [←]. The number you see will be in the units that are in the ALPHA register. Example 1 of the *Water Viscosity* program illustrates this.

English or SI Default Units Using Flag 09

Table 1 shows two sets of default units, English and SI (the International System of Units). The Pac allows you to select the unit system you use the most often with flag 09. If you set flag 09 ([SF] 09), all input and output prompts will be in SI default units. If you clear flag 09 ([CF] 09), all input and output prompts will be in English default units.

Like flag 10, **the status of flag 09 (set or clear) is remembered by the HP-41 when it is turned off, and is never altered by any program in the Pac.** This means you only have to set or clear the flag once, and the default units will always be in the desired unit system whenever you use the Pac. You may want to use units that are not the English or SI default units. The available units are listed in Table 3.

Example 1 of the *Oil Isothermal Compressibility* program illustrates the use of flag 09 for SI default units.

Shown on the next page is a table to summarize the relationship between flags 09 and 10.

Table 2: Relationship Between Flags 09 and 10

	Flag 10 Clear	Flag 10 Set
Flag 09 Clear	Pac English default units will be used on input and output. Program will not halt on output—English units will be automatically selected.	Pac English default units will be used on input and output. Program will beep, stop, and prompt you for output units.
Flag 09 Set	Pac SI default units will be used on input and output. Program will not halt on output—SI units will be automatically selected.	Pac SI default units will be used on input and output. Program will beep, stop, and prompt you for output units.

Number of Characters Allowed for Units

When using the programs in the Petroleum Fluids Pac, you may key in no more than 12 characters for units in response to an input or output prompt.

Output Units Saved

In addition to providing default units for input and output variables, the programs in the Pac also save the units of the primary output variable. Once you have selected the output units you want, you won't have to reenter them when the prompt reappears.

The primary output variable is the one the program was named after. In the *Oil Viscosity* program, for example, GAS GS, PBP, and RSb are output in addition to the viscosity, but only the viscosity units will be saved.

The output units are saved for all programs except *Pseudocritical Temperature and Pressure From Gas Gravity*. In the *Gas Properties From Composition* program, the output units for either the heating values or for the specific heats are saved, depending on the response to the ***SP.HTS? Y/N*** question.

Example 1 of the *Water Viscosity* program illustrates how output units are saved.

Invalid Units

The unit abbreviations you key in are checked for validity. If you key in the units of a variable incorrectly, the HP-41 will give you another chance. If you make an error following an input prompt, the letters you keyed in followed by a question mark will be displayed. For the previous input example, the input prompt for pressure was:

P=?

If you had responded with 103 [ALPHA] CP [R/S] (viscosity), the following would have appeared:

CP?

If you make an error following an output prompt, the prompt will be repeated, followed by the letters you keyed in instead of the original units. For the previous output example, the output prompt for gas formation volume factor was:

BG, FT3/SCF?

If you had responded with R [R/S] (degrees Rankine), the following would have appeared:

BG, R?

In both cases, ALPHA mode will be on to allow you to change the incorrect units. Simply key in the correct units and press [R/S]. Errors made in specifying units fall into the following four categories:

1. The unit control characters (*, /, −, 1-9) were used incorrectly (e.g., FT/S/S). These will be discussed shortly.
2. More than 12 characters were used to specify the units.
3. The units specified were incompatible with the requested input or output variable (e.g., the prompt was for pressure and the units you supplied were for viscosity).
4. The abbreviations used were not in the list of basic units shown in Table 3. Note that no lowercase letters are allowed.

Note: There are a few instances in which the units are valid, but the number in X is not, specifically when converting either zero or −131.5 API. For these cases, after the input or output error prompt appears, press [ALPHA] and key in a valid number. Then press [R/S] to continue running the program.

Table 3: Petroleum Engineering Basic Units

HP-41 Abbreviation	Name	Multiplicative Conversion Constant	Homogenous SI Unit
ACRE	acre	$4.046856422 \times 10^{3}$	m^2
API	degree API*	$1.413643345 \times 10^{5}$/ (API + 131.5)	kg/m^3
ATM	atmosphere	1.01325×10^{5}	Pa
BAR	bar	1.0×10^{5}	Pa
BBL	barrel of petroleum	$1.589872949 \times 10^{-1}$	m^3
BCF	billion standard cubic feet of gas‡	1.1953×10^{6}	kg·mol
BTU	British Thermal Unit (IST)†	1.055056×10^{3}	J
C	degree Celsius	1.0×10^{0} (+ 273.15)	K
CAL	calorie (IST)†	4.1868×10^{0}	J
CM	centimeter	1.0×10^{-2}	m
CP	centipoise	1.0×10^{-3}	Pa·s
CST	centistoke	1.0×10^{-6}	m^2/s
D	darcy	9.869233×10^{-13}	m^2
DAY	day	8.64×10^{4}	s
DYNE	dyne	1.0×10^{-5}	N
ERG	erg	1.0×10^{-7}	J
F	degree Fahrenheit	$(F + 459.67) \times 5.555555555 \times 10^{-1}$	K
FT	foot	3.048×10^{-1}	m
FTH2O	foot of water (39.2 F)	2.98898×10^{3}	Pa
G	gram	1.0×10^{-3}	kg
GAL	gallon (U.S.)	$3.785411784 \times 10^{-3}$	m^3
GALUK	gallon (U.K.)	4.546087×10^{-3}	m^3
HP	horsepower (550 ft*lbf/s)	7.4569987×10^{2}	W
HR	hour (mean solar)	3.6×10^{3}	s
IN	inch	2.54×10^{-2}	m
INHG	inch of mercury (60 F)	3.37685×10^{3}	Pa
INH2O	inch of water (60 F)	2.4884×10^{2}	Pa
J	joule	1.0×10^{0}	J

Due to the additive nature of the API conversion, API may not appear with other units using "" or "/".

†International Steam Table.

‡The units of volume at standard conditions are equivalent to an amount of substance, not to a length cubed. Therefore, SCF, SCM, SCMZ, and related units are not dimensionally consistent with other volume units such as FT3, M3, BBL, etc.

HP-41 Abbreviation	Name	Multiplicative Conversion Constant	Homogenous SI Unit
K	Kelvin	1.0×10^{0}	K
KCAL	kilocalorie (IST)†	4.1868×10^{3}	J
KG	kilogram	1.0×10^{0}	kg
KGF	kilogram force	9.80665×10^{0}	N
KIP	kilopound force	$4.448221615 \times 10^{3}$	N
KJ	kilojoule	1.0×10^{3}	J
KM	kilometer	1.0×10^{3}	m
KMOL	kilomole	1.0×10^{3}	mol
KPA	kilopascal	1.0×10^{3}	Pa
KSI	kip per square inch	6.8947572×10^{6}	Pa
KT	kilotonne	1.0×10^{6}	kg
KW	kilowatt	1.0×10^{3}	W
L	liter	1.0×10^{-3}	m^3
LBF	pound force	$4.448221615 \times 10^{0}$	N
LBM	pound mass	4.5359237×10^{-1}	kg
M	meter	1.0×10^{0}	m
MBAR	millibar	1.0×10^{2}	Pa
MCF	thousand standard cubic feet of gas‡	1.1953×10^{0}	kg·mol
MD	millidarcy	9.869233×10^{-16}	m^2
MG	megagram	1.0×10^{3}	kg
MI	mile	1.609344×10^{3}	m
MIN	minute	6.0×10^{1}	s
MJ	megajoule	1.0×10^{6}	J
ML	milliliter	1.0×10^{-6}	m^3
MM	millimeter	1.0×10^{-3}	m
MMCF	million standard cubic feet of gas‡	1.1953×10^{3}	kg·mol
MMHG	millimeter of mercury (60 F)	1.333224×10^{2}	Pa
MN	meganewton	1.0×10^{6}	N
MO	month	2.628×10^{6}	s
MOL	mole	1.0×10^{0}	mol
MPA	megapascal	1.0×10^{6}	Pa
MT	megatonne	1.0×10^{9}	kg
MW	megawatt	1.0×10^{6}	W
N	newton	1.0×10^{0}	N
P	poise	1.0×10^{-1}	Pa·s
PA	pascal	1.0×10^{0}	Pa
PSF	pound force per square foot	$4.788025898 \times 10^{1}$	Pa

†International Steam Table.

‡The units of volume at standard conditions are equivalent to an amount of substance, not to a length cubed. Therefore, SCF, SCM, SCMZ, and related units are not dimensionally consistent with other volume units such as FT3, M3, BBL, etc.

HP-41 Abbreviation	Name	Multiplicative Conversion Constant	Homogenous SI Unit
PSI	pound force per square inch	$6.894757293 \times 10^{3}$	Pa
R	degree Rankine	$5.555555555 \times 10^{-1}$	K
S	second	1.0×10^{0}	s
SCF	standard cubic foot (60 F, 14.696 psi)‡	1.1953×10^{-3}	kg·mol
SCM	standard cubic meter (15 C, 101.325 kPa)‡	4.22932×10^{-2}	kg·mol
SCMZ	standard cubic meter (0 C, 101.325 kPa)‡	4.46158×10^{-2}	kg·mol
SPGR	specific gravity relative to water (60 F)	$9.990412333 \times 10^{2}$	kg/m^3
ST	stoke	1.0×10^{-4}	m^2/s
T	tonne (metric ton, 1000 kg)	1.0×10^{3}	kg
THERM	10^5 Btu	1.055056×10^{8}	J
TON	short ton (2000 lbm)	9.0718474×10^{2}	kg
TONUK	long ton (2240 lbm)	$1.016046909 \times 10^{3}$	kg
TORR	torr (0 C)	1.33322×10^{2}	Pa
UM	micrometer	1.0×10^{-6}	m
W	watt	1.0×10^{0}	W
YD	yard	9.144×10^{-1}	m
YR	year (calendar)	3.1536×10^{7}	s
	null string	1.0×10^{0}	

‡The units of volume at standard conditions are equivalent to an amount of substance, not to a length cubed. Therefore, SCF, SCM, SCMZ, and related units are not dimensionally consistent with other volume units such as FT3, M3, BBL, etc.

A Comment on Terminology

The "homogenous SI units" are the SI equivalents of each basic unit in Table 3. The "Pac default units" are the combinations of these basic units expected by the different variables in the Pac. Table 1 lists the Pac default units used by each general category of variables (such as all the pressures, all the viscosities, etc.) Appendix C is a list of all the specific variables used in the Pac (such as separator pressure, dead oil viscosity, etc.) and the Pac default units for these variables.

Using the Unit Control Characters

The basic units from Table 3 may be combined into unit strings using the unit control characters (*, /, −, 1-9). The "*" means multiply, and the "/" means divide. Digits 1-9 indicate the power to which a unit should be

raised. Only single digits are used as exponents. For instance, an acceleration in feet per second squared would be keyed in as:

*FT/S*S*

or

FT/S2

Legal unit strings for volumetric flow rate include, but are far from limited to:

FT3/S
*M*CM*IN/MIN*
*FT3*HR/S2*

etc.

Only one divide sign is allowed in the unit string. Thus, all units to the right of the divide sign are included in the denominator.

In response to an input or output prompt, you can use any combination of units from Table 3 that are dimensionally consistent with the default units for that variable.

Using [CON] and [INCON]

There will probably be occasions when you wish to use the Unit Management System directly from the keyboard. Two functions, [CON] and [INCON], form the basis of the system. [CON] means conversion, and [INCON] means inverse conversion. Both functions look at the characters in the ALPHA register, decide, based on these characters, what conversions need to be done, and perform these conversions on the value in the X register.

The characters keyed into the ALPHA register form a unit equation. They tell the HP-41 the units of the value in X and the units to which the value should be converted. For instance, to convert from feet to inches, the unit equation would be as follows:

FT-IN

Notice that the dash or minus sign is read "converted to." Therefore, the unit equation says "feet converted to inches." If we execute [CON] with this unit equation in the ALPHA register, the value in X will be converted from feet to inches. If we execute [INCON], the inverse conversion will be performed, and the value in X will be converted from inches to feet. In either case, LAST X will contain the original value of X, the Y, Z, and T registers will be unaffected, and the ALPHA register will be unchanged.

A shortcut is available if you wish to convert to or from SI units. In such cases, you do not need to specify the right hand side of the unit equation. To convert from feet to meters, for example, simply key "FT" into the

ALPHA register and execute [CON]. Since the meter is the SI unit of length, it will be assumed for the right side of the unit equation. Table 3 defines the homogenous SI units used in the Pac.

Number of Characters Allowed in Unit Equation

When using [CON] and [INCON], you may key in any combination of alphabetic and unit control characters (*, /, −, 1-9) up to the limit of the ALPHA register (24 characters). Note that this is different than when you are using a **program** in the Petroleum Fluids Pac, in which you are limited to 12 characters.

User Instructions for [CON] and [INCON]

				SIZE: any
STEP	**INSTRUCTIONS**	**INPUT**	**FUNCTION**	**DISPLAY**
1	Key in unit equation.	Unit Equation	[ALPHA] [ALPHA]	
2	Key in numeric value to be converted.	Value		
3a	Perform conversion.		[XEQ] CON	Converted Value
3b	Perform inverse conversion.		[XEQ] INCON	Converted Value
4	For a new case, go to steps 1 or 2. The original value is in LAST X. The Y, Z, T, and ALPHA registers have not been altered. Notice that steps 1 and 2 do not have to be done in order.			

Note: Because these two functions are written in machine language, you will not be able to copy, list, or single step through them.

Example 1:

Convert 212 F to degrees Celsius. Convert 0 C to degrees Fahrenheit.

Keystrokes	Display	
[ALPHA] F-C [ALPHA]		
212 [XEQ] [ALPHA] CON [ALPHA]	*100.0000*	C
0 [XEQ] [ALPHA] INCON [ALPHA]	*32.0000*	F

Example 2:

Convert 23 pounds per square inch to atmospheres.

Keystrokes	Display	
23 [ALPHA] PSI-ATM [ALPHA]		
[XEQ] [ALPHA] CON [ALPHA]	*1.5651*	ATM

Example 3:

Convert 88 feet per second to kilometers per hour. Convert 100 kilometers per hour back to feet per second.

Keystrokes	Display	
[ALPHA] FT/S-KM/HR [ALPHA]		
88 [XEQ] [ALPHA] CON [ALPHA]	*96.5606*	KM/HR
100 [XEQ] [ALPHA] INCON [ALPHA]	*91.1344*	FT/S

Example 4:

Using the short form SI conversion, convert 10 feet to meters and 4 meters to feet.

Keystrokes	Display	
10 [ALPHA] FT [ALPHA]		
[XEQ] [ALPHA] CON [ALPHA]	*3.0480*	M
4 [XEQ] [ALPHA] INCON [ALPHA]	*13.1234*	FT

Example 5:

Perform the following unit conversion:

$$20\ \frac{(\text{BTU})\,(\text{IN})}{(\text{FT}^3)\,(\text{F})\,(\text{S})}\ \text{to}\ \frac{\text{W}}{(\text{IN}^2)\,(\text{C})}$$

Keystrokes	Display	
[ALPHA] BTU*IN/FT3*F*S-		
W/IN2*C [ALPHA]		
20 [XEQ] [ALPHA] CON [ALPHA]	*21.9803*	W/IN2*C

Example 6:

Find the specific gravity of 36 degree API oil.

Keystrokes	Display	
36 [ALPHA] API-SPGR [ALPHA] [XEQ] [ALPHA] CON [ALPHA]	***0.8448***	SPGR

[CON] and [INCON] Error Conditions

There are several error conditions which will prevent a unit conversion from taking place. If any of these conditions is encountered while using [CON] or [INCON], the error message ***INVALID CONV*** (invalid conversion) will be displayed. This error message behaves like any other HP-41 error message, such as ***DATA ERROR***, etc. If the message occurs, no registers will have been altered. If flag 25 had been set, it will now be clear. The error conditions are listed below.

1. ALPHA data in X.
2. Invalid character in unit equation (not *, /, −, 1-9, or A-Z).
3. Zero exponent on unit; e.g., FT0.
4. Multiple dashes in unit equation.
5. Multiple "/" on either side of unit equation.
6. Dimensional inconsistency.
7. Unit not in Table 2.
8. API with either "*" or "/".
9. With API (or API-unit) in ALPHA, attempting [CON] with −131.5 in X or [INCON] with 0 in X.
10. With unit-API in ALPHA, attempting [CON] with 0 in X or [INCON] with −131.5 in X.

Notes

Z FACTOR

The Z program estimates the real gas deviation, or Z factor (Z), for hydrocarbon gases. The technique used by this program was developed by Dranchuk, Purvis, and Robinson. The Benedict-Webb-Rubin equation of state was fit to the Z factor surface defined by the Standing-Katz Z factor correlation. The resulting nonlinear equation is then solved for Z as a function of reduced temperature and pressure using Newton-Raphson iteration.

Equations:

$$Z = \frac{0.27\,PR}{\rho_r TR}$$

The pseudoreduced density, ρ_r, is found iteratively:

$$\rho_{r_{i+1}} = \rho_{r_i} - \frac{f(\rho_r)}{f'(\rho_r)}$$

$$f(\rho_r) = A\,\rho_r^6 + B\,\rho_r^3 + C\,\rho_r^2 + D\,\rho_r + E\,\rho_r^3(1 + F\,\rho_r^2)\exp(-F\,\rho_r^2) - G$$

$$f'(\rho_r) = 6A\,\rho_r^5 + 3B\,\rho_r^2 + 2C\,\rho_r + D + E\,\rho_r^2[3 + F\,\rho_r^2(3 - 2F\,\rho_r^2)]\exp(-F\,\rho_r^2)$$

$$A = 0.06423$$

$$B = 0.5353\,TR - 0.6123$$

$$C = 0.3151\,TR - 1.0467 - \frac{0.5783}{TR^2}$$

$$D = TR$$

$$E = \frac{0.6816}{TR^2}$$

$$F = 0.6845$$

$$G = 0.27\,PR$$

$$\rho_{r_0} = \frac{0.27\,PR}{TR}\ \text{(initial guess)}$$

Calculation Subroutine:

The calculation subroutine for this program is called CZ (*Calculate Z*). Since it is written in machine language, you will not be able to copy, list, or single step through it. The routine expects TR in Y and PR in X. Upon return, Z will be in X, TR will be in Y, PR will be in LAST X, and the Z, T, and ALPHA registers will be cleared.

Range of Validity:

The program will halt and display ***INVALID TR*** for reduced temperatures outside the following range:

$$1.05 \leqslant \mathrm{TR} < 3.0$$

If this occurs, press [←] [$x \gtrless y$] to see TR. Then restart the program, using a different T or Tc so that the ratio TR = T/Tc will be within the above range.

You should be aware that for temperature isotherms corresponding to reduced temperatures below 1.2, both the algorithm and the experimental data upon which it is based are of questionable validity. If the accuracy is a significant factor to you, you may want to compute TR to verify that it is in an acceptable range.

The program will halt and display ***INVALID PR*** for pressures outside the following range:

$$0 < \mathrm{PR} < 30$$

If this occurs, press [←] to see PR. Then restart the program, using a different P or Pc so that the ratio PR = P/Pc will be within the above range.

Remarks:

The Dranchuk, Purvis, and Robinson correlation has the lowest average error of the eight most commonly used methods for estimating Z factors. See the second reference for more details.

References:

Dranchuk, P. M., Purvis, R. A., and Robinson, D. B., "Computer Calculations of Natural Gas Compressibility Factors Using the Standing and Katz Correlation," *Institute of Petroleum Technical Series,* No. IP 74-008, 1974.

Takacs, G., "Comparisons Made for Computer Z-Factor Calculations," *Oil and Gas Journal,* Dec. 20, 1976, pp. 64-66.

User Instructions:

				SIZE: 018
STEP	**INSTRUCTIONS**	**INPUT**	**FUNCTION**	**DISPLAY**
1	Initialize program.		XEQ Z	Tc=?
2	Key in pseudocritical temperature.	Tc	R/S	Pc=?
3	Key in pseudocritical pressure.	Pc	R/S	T=?
4	Key in temperature.	T	R/S	P=?
5	Key in pressure and calculate Z.	P	R/S R/S*	Z= P=?
6	For a new pressure, go to step 5. For a new case, go to step 1. Key in only values which change. For values which do not change, ignore the prompts and press R/S. * Press R/S if you are not using a printer.			

Example 1:

A gas has a pseudocritical temperature of 383 R and a pseudocritical pressure of 45.4 ATM. At a temperature of 150 C and a pressure of 6340 PSI, what is the Z factor for the gas? Start with size 000 to observe the program's response when not enough data registers are available.

Keystrokes	**Display**
XEQ ALPHA SIZE ALPHA 000	
XEQ ALPHA Z ALPHA	*SIZE>=18.0000*
XEQ ALPHA SIZE ALPHA 018	
R/S	*Tc=?*
383 ALPHA	*R*

The ALPHA key was pressed to check the default units for pseudocritical temperature. Since R is the abbreviation for degrees Rankine, and the input is in degrees Rankine, complete the input of temperature by pressing R/S (it is not necessary to press the ALPHA key again).

R/S	*Pc=?*

For pseudocritical pressure, the Pac default units are PSI. Since the given value is in atmospheres, the abbreviation for atmospheres must be input in ALPHA mode.

45.4 ALPHA	*PSI*
ATM R/S	*T=?*

Similarly, since the Pac default units for temperature are degrees F, and the given value is in degrees C, the abbreviation for degrees C must be input in ALPHA mode.

150 [ALPHA]	*F*
C [R/S]	*P=?*

Since the default units for pressure are PSI, and the given pressure is in PSI, simply key in the number and press [R/S].

6340 [R/S]	*Z=1.1248*

```
Z FACTOR

Tc=383.0000 R
Pc=45.4000 ATM
T=150.0000 C
P=6340.0000 PSI

Z=1.1248
```

For programmers who want to use Pac calculation subroutines in their own programs, the following example illustrates the use of these routines. If you do not intend to use these subroutines in your programs, you should skip the next example.

Example 2:

For a reduced temperature of 1.5 and a reduced pressure of 5.0, use the calculation subroutine to compute the Z factor.

Keystrokes	Display	
1.5 [ENTER↑] 5		
[XEQ] [ALPHA] CZ [ALPHA]	*0.8110*	Z
[R↓]	*1.5000*	TR
[R↓]	*0.0000*	
[R↓]	*0.0000*	
[LASTx]	*5.0000*	PR

Notice that TR remained in Y, PR was placed in LAST X, and the rest of the stack was cleared.

GAS ISOTHERMAL COMPRESSIBILITY

The CG program estimates the isothermal coefficient of compressibility (CG) for hydrocarbon gases. The Z factor technique developed by Dranchuk, Purvis, and Robinson was adapted to compute the partial derivative of Z with respect to pressure at constant temperature. This derivative is used to compute the pseudoreduced compressibility (CR). CG is then calculated from CR using the technique developed by Trube.

Equations:

$$CG = \frac{CR}{Pc}$$

$$CR = \frac{1}{PR} \frac{1}{1 + (\rho_r/Z)(\partial Z/\partial \rho_r)}$$

The pseudoreduced density, ρ_r, is found iteratively using the algorithm shown in the "Equations" section of the *Z Factor* program. Coefficients A, B, C, E, and F from that section are used with the calculated value of ρ_r to compute $\partial Z/\partial \rho_r$ as follows:

$$\frac{\partial Z}{\partial \rho_r} = \frac{1}{\rho_r TR}[5A\ \rho_r^5 + 2B\ \rho_r^2 + C\ \rho_r + 2E\ \rho_r^2 (1 + F\ \rho_r^2 - F^2 \rho_r^4) \exp(-F\ \rho_r^2)]$$

Calculation Subroutines:

There are two calculation subroutines for this program. The first subroutine is called CCG (*Calculate CG*). The routine expects Pc in the proper register (see Appendix B). In addition, TR must be in Y and PR must be in X. Upon return, CG in 1/PSI will be in X.

The second calculation subroutine is called CCR (*Calculate CR*). Since it is written in machine language, you will not be able to copy, list, or single step through it. The routine expects TR in Y and PR in X. Upon return, CR will be in X, TR will be in Y, PR will be in LAST X, and the Z, T, and ALPHA registers will be cleared.

Range of Validity:

See the "Range of Validity" section of the *Z Factor* program.

References:

Meehan, D. N., and Lyons, W. K., "Calculations Programmable for Gas Compressibility," *Oil and Gas Journal,* Oct. 8, 1979, pp. 74-78.

Trube, A. S., "Compressibility of Natural Gases," *Trans. AIME,* 210, 1957, p. 335.

User Instructions:

				SIZE: 018
STEP	**INSTRUCTIONS**	**INPUT**	**FUNCTION**	**DISPLAY**
1	Initialize program.		XEQ CG	Tc=?
2	Key in pseudocritical temperature.	Tc	R/S	Pc=?
3	Key in pseudocritical pressure.	Pc	R/S	T=?
4	Key in temperature.	T	R/S	P=?
5	Key in pressure and calculate CG.	P	R/S R/S*	CG=† P=?
6	For a new pressure, go to step 5. For a new case, go to step 1. Key in only values which change. For values which do not change, ignore the prompts and press R/S. *Press R/S if you are not using a printer. †Press ALPHA to see the units if you are not using a printer.			

Example 1:

Compute the isothermal coefficient of compressibility for a gas with Pc of 39.1 ATM and Tc of 383 R. The actual pressure is 335 ATM and the actual temperature is 153 F. What is the isothermal compressibility for the gas in 1/KPA? Turn on the output portion of the Unit Management System (set flag 10) to allow you to change the output units.

Keystrokes (SIZE >= 018)	**Display**	
SF 10		
XEQ ALPHA CG ALPHA	*Tc=?*	
383 R/S	*Pc=?*	
39.1 ALPHA ATM R/S	*T=?*	
153 R/S	*P=?*	
335 ALPHA ATM R/S	*CG, 1/PSI?*	
1/KPA R/S	*CG=1.4460E-5*	1/KPA†
CF 10		

†Press ALPHA to see the units if you are not using a printer.

```
GAS ISO CMP

Tc=383.0000 R
Pc=39.1000 ATM
T=153.0000 F
P=335.0000 ATM

CG=1.4460E-5 1/KPA
```

For programmers who want to use Pac input and calculation subroutines in their own programs, the following examples illustrate the use of these routines. If you do not intend to use these subroutines in your programs, you should skip the next examples.

Example 2:

For a reduced temperature of 1.7, a reduced pressure of 2.8, and a Pc of 361.5 PSI, use the calculation subroutine to compute the isothermal compressibility in 1/PSI. First, use the pseudocritical temperature and pressure input routine (ITcPc) to input and store the new Pc. (Note that this input subroutine is called ITcPc to avoid conflict with the TcPc program.)

Keystrokes	Display	
XEQ ALPHA ITcPc ALPHA	*Tc=?*	
R/S	*Pc=?*	
361.5 R/S	*361.5000*	Pc (PSI)

Now compute the compressibility.

1.7 ENTER↑ 2.8		
XEQ ALPHA CCG ALPHA	*0.0010*	CG (1/PSI)
	Pc=361.5000 PSI	

Example 3:

For a reduced temperature of 1.5 and a reduced pressure of 5.0, use the second calculation subroutine to compute the pseudoreduced compressibility.

Keystrokes	Display	
1.5 ENTER↑ 5		
XEQ ALPHA CCR ALPHA	*0.1459*	CR
R↓	*1.5000*	TR

Keystrokes	Display	
R↓	*0.0000*	
R↓	*0.0000*	
f LAST*x*	*5.0000*	PR

Notice that TR remained in Y, PR was placed in LAST X, and the rest of the stack was cleared.

GAS FORMATION VOLUME FACTOR

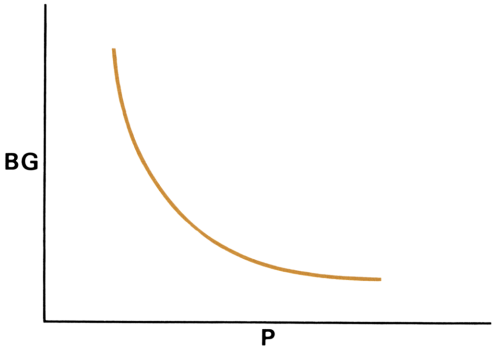
The BG program estimates the formation volume factor (BG) for reservoir gases. For use in the Pac, BG is defined as the ratio of the volume at reservoir conditions to the volume at standard conditions. The volume at reservoir conditions is calculated using the ideal gas law with Z factor correction.

Equations:

$$BG = \frac{Z\,T'\,STD\,P}{P\,STD\,T'}$$

$$T' = T \text{ in } R$$

$$STD\,T' = STD\,T \text{ in } R$$



Calculation Subroutine:

The calculation subroutine for this program is called CBG (*Calculate BG*). The routine expects certain variables (Pc, STD T, STD P, T) in the proper registers (see Appendix B). In addition, TR must be in Y and PR must be in X. Upon return, BG in FT3/SCF will be in X.

Range of Validity:

See the "Range of Validity" section of the *Z Factor* program.

Remarks:

This program is one of several that prompts you for the temperature and pressure at standard conditions (STD T and STD P). If the previously stored value of STD T is 0 F (i.e., if the registers were cleared), the program will automatically replace that value with 60 F before the ***STD T=?*** prompt appears. If the previously stored value of STD P is 0 PSI, the program will automatically replace that value with 14.65 PSI before the ***STD P=?*** prompt appears.

60 F is a universal value for STD T (except in Canada, where the standard is 15 C), but STD P varies depending upon location. Table 4 lists the standard pressures used in various locations.

Table 4: Standard Pressures by Location

Location	STD P, PSI
Arkansas	14.65
California	14.73
Colorado	15.025
Illinois	14.65
Kansas	14.65
Louisiana	15.025
Michigan	14.73
Mississippi	15.025
New Mexico	15.025
Oklahoma	14.65
Texas	14.65
Utah	15.025
West Virginia	14.85
Wyoming	15.025
U.S. Federal Leases	14.73
Canada	14.696**

**101.325 KPA

References:

McCain, W. D., *The Properties of Petroleum Fluids*, Petroleum Publishing Co., 1973, pp. 122-124.

Amyx, Bass, and Whiting, *Petroleum Reservoir Engineering*, McGraw-Hill, 1968, p. 30.

User Instructions:

				SIZE: 024
STEP	**INSTRUCTIONS**	**INPUT**	**FUNCTION**	**DISPLAY**
1	Initialize program.		XEQ BG	Tc=?
2	Key in pseudocritical temperature.	Tc	R/S	Pc=?
3	Key in pseudocritical pressure.	Pc	R/S	STD T=?
4	Key in temperature at standard conditions. If the previously stored STD T is 0 F, it will be replaced by 60 F before this prompt appears.	STD T	R/S	STD P=?
5	Key in pressure at standard conditions. If the previously stored STD P is 0 PSI, it will be replaced by 14.65 PSI before this prompt appears.	STD P	R/S	T=?
6	Key in temperature.	T	R/S	P=?
7	Key in pressure and calculate BG.	P	R/S R/S*	BG=† P=?
8	For a new pressure, go to step 7. For a new case, go to step 1. Key in only values which change. For values which do not change, ignore the prompts and press R/S. *Press R/S if you are not using a printer. †Press ALPHA to see the units if you are not using a printer.			

Example 1:

A gas has Tc and Pc of 390 R and 670 PSI, respectively. For reservoir conditions of 205 F and 1500 PSI, what is BG in FT3/SCF? Initialize STD T and STD P to 0 F and 0 PSI, respectively, by clearing all the data registers. Then use the automatically assigned values of STD T and STD P.

Keystrokes (SIZE>=024)	Display
XEQ ALPHA CLRG ALPHA	
XEQ ALPHA BG ALPHA	*Tc=?*
390 R/S	*Pc=?*
670 R/S	*STD T=?*
←	*60.0000*
ALPHA	*F*

Since the previously stored value of STD T was 0 F because the registers were cleared, it was replaced by 60 F before the prompt appeared.

R/S	*STD P=?*
←	*14.6500*
ALPHA	*PSI*

Since the previously stored value of STD P was 0 PSI because the registers were cleared, it was replaced by 14.65 PSI before the prompt appeared.

R/S	*T=?*	
205 R/S	*P=?*	
1500 R/S	*BG=0.0110*	FT3/SCF†

```
GAS VOL FACT

Tc=390.0000 R
Pc=670.0000 PSI
T=205.0000 F
P=1500.0000 PSI

BG=0.0110 FT3/SCF
```

The units used for BG must be a volume unit in the numerator (FT3, M3, BBL, etc.) and a volume at standard conditions unit in the denominator (SCF, SCM, etc.). The reciprocal of this definition for BG is commonly used in industry, but is not allowed by the BG program. This is because the Unit Management System treats the units of volume at standard conditions as an amount of substance, not as a length cubed. Therefore, FT3/SCF is not dimensionally consistent with SCF/FT3. (Refer to *The Petroleum Engineering Unit Management System* for further details.)

For programmers who want to use Pac input and calculation subroutines in their own programs, the following example illustrates the use of these routines. If you do not intend to use these subroutines in your programs, you should skip the next example.

†Press ALPHA to see the units if you are not using a printer.

Example 2:

Assuming you have just run the last example, use the calculation subroutine to compute the formation volume factor in FT3/SCF for the same gas at 400 K. Use the temperature input routine (T) to input and store the new T.

Keystrokes	Display	
XEQ ALPHA T ALPHA	*T=?*	
400 ALPHA K R/S	*260.3300*	T (F)

(Remember that all input and calculation subroutines return values in Pac English default units.) The calculation subroutine requires the reduced temperature in Y and the reduced pressure in X. TR is calculated as T/Tc, where T and Tc must be in the same units. Since Tc is in R, convert T to R and calculate TR. The temperature just input is already in X in Pac English default units (F).

ALPHA F-R ALPHA	*260.3300*	T (F)
XEQ ALPHA CON ALPHA	*720.0000*	T (R)
390 ÷	*1.8462*	TR

Next, calculate PR as P/Pc, where P and Pc must be in the same units.

1500 ENTER↑ 670 ÷	*2.2388*	PR

Finally, with TR in Y and PR in X, calculate the formation volume factor.

XEQ ALPHA CBG ALPHA	*0.0124*	BG (FT3/SCF)
	T=400.0000 K	

Notes

GAS VISCOSITY

The UG program estimates the dynamic gas viscosity (UG) for hydrocarbon gases. The calculation is based on an estimate of the density of the gas, which is calculated using the ideal gas law with Z factor correction.

There is a correction for sour gas content inherent in the pseudocritical properties. If Tc and Pc have been corrected for sour gases, that correction will propagate through the Z factor calculation to the viscosity. (Both Pac programs that compute pseudocritical properties correct Tc and Pc for sour gas content.)

Equations:

$$UG = A \exp(B\, \rho'^{C})$$

$$A = \frac{(9.4 + 0.02\,MW)\,T'^{1.5}}{(209 + 19\,MW + T')\,10^4}$$

$$B = 3.5 + \frac{986}{T'} + 0.01\,MW$$

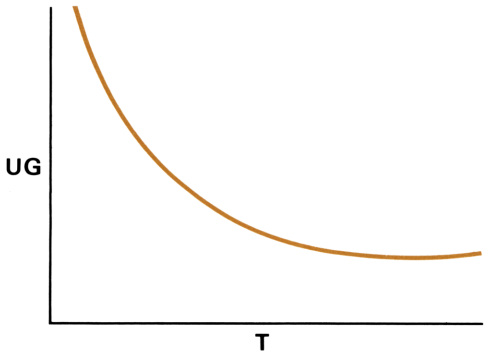
$$C = 2.4 - 0.2\,B$$

$$\rho' = \frac{P\,MW}{Z\,\boldsymbol{R}\,T'}$$

$\rho' = \rho$ (density) in G/CM3

$T' = T$ in R

$$\boldsymbol{R} = \text{universal gas constant} = 669.8\,\frac{CM3 \cdot PSI}{G \cdot MOL \cdot R}$$



Calculation Subroutine:

The calculation subroutine for this program is called CUG (*Calculate UG*) The routine expects certain variables (GAS G, T, P) in the proper registers (see Appendix B), and uses certain scratch registers (see Appendix A). In addition, TR must be in Y and PR must be in X. Upon return, UG in CP will be in X.

Range of Validity:

$$40 < T < 460 \text{ F}$$
$$14.7 < P < 10000 \text{ PSI}$$

See also the "Range of Validity" section of the *Z Factor* program.

References:

Lee, A. L., Gonzalez, M. H., and Eakin, B. E., "The Viscosity of Natural Gases," *Journal of Petroleum Technology,* August, 1966, pp. 997-1000.

Gonzalez, M. H., and Lee, A. L., "Graphical Viscosity Correlation for Hydrocarbons," *AIChE Journal,* March, 1968, pp. 242-244.

User Instructions:

				SIZE: 018
STEP	**INSTRUCTIONS**	**INPUT**	**FUNCTION**	**DISPLAY**
1	Initialize program.		XEQ UG	Tc=?
2	Key in pseudocritical temperature.	Tc	R/S	Pc=?
3	Key in pseudocritical pressure.	Pc	R/S	GAS G=?
4	Key in gas gravity. If you do not know gas gravity, but do know molecular weight, press R/S to get the molecular weight prompt, and then key in molecular weight.	GAS G or MW	R/S R/S R/S	T=? MW=? T=?
5	Key in temperature.	T	R/S	P=?
6	Key in pressure and calculate UG.	P	R/S R/S*	UG=† P=?
7	For a new pressure, go to step 6. For a new case, go to step 1. Key in only values which change. For values which do not change, ignore the prompts and press R/S.			
	* Press R/S if you are not using a printer.			
	†Press ALPHA to see the units if you are not using a printer.			

Example 1:

Natural gas containing 2 percent nitrogen, 1 percent carbon dioxide, and 7 percent hydrogen sulfide has a gravity of 0.74. Determine the viscosity of the gas in CP at 618 R and 125 ATM.

The pseudocritical temperature and pressure for this viscosity calculation may be obtained using the *Pseudocritical Temperature and Pressure From Gas Gravity* program (TcPc) elsewhere in the Pac. (Note: answer N to the ***COND? Y/N*** question.) The pseudocritical properties calculated by that program are 391 R for Tc and 688 PSI for Pc.

Keystrokes (SIZE >= 018)	**Display**	
XEQ ALPHA UG ALPHA	***Tc=?***	
391 R/S	***Pc=?***	
688 R/S	***GAS G=?***	
.74 R/S	***T=?***	
618 ALPHA R R/S	***P=?***	
125 ALPHA ATM R/S	***UG=0.0164***	CP†

```
GAS VIS

Tc=391.0000 R
Pc=688.0000 PSI
GAS G=0.7400
T=618.0000 R
P=125.0000 ATM

UG=0.0164 CP
```

For programmers who want to use Pac input and calculation subroutines in their own programs, the following example illustrates the use of these routines. If you do not intend to use these subroutines in your programs, you should skip the next example.

Example 2:

Assuming you have just run the last example, use the calculation subroutine to compute the viscosity in CP for the same gas at 250 ATM. Input and store the new P using the pressure input routine (P).

†Press ALPHA to see the units if you are not using a printer.

Keystrokes	Display	
XEQ ALPHA P ALPHA	*P=?*	
250 ALPHA ATM R/S	*3673.9872*	P (PSI)

(Remember that all input and calculation subroutines return values in Pac English default units.) The pressure input routine returns with TR in Z, PR in Y, and P in X.

R↓	*5.3401*	PR

With TR in Y and PR in X, calculate the viscosity.

XEQ ALPHA CUG ALPHA	*0.0248*	UG (CP)
	P=250.0000 ATM	

PSEUDOCRITICAL TEMPERATURE AND PRESSURE FROM GAS GRAVITY

The TcPc program estimates the pseudocritical temperature (Tc) and pseudocritical pressure (Pc) from gas gravity for both condensate fluids and miscellaneous reservoir gases. The calculated values for Tc and Pc are corrected for sour gas content (Tc* and Pc*) with a Wichert-Aziz correction (CWA).

Equations:

Condensate Fluids:

$$Tc_{HC} = 187 + 330\ GAS\ G_{HC} - 71.5\ GAS\ G_{HC}^2$$

$$Pc_{HC} = 706 - 51.7\ GAS\ G_{HC} - 11.1\ GAS\ G_{HC}^2$$

Miscellaneous Gases:

$$Tc_{HC} = 168.0 + 325\ GAS\ G_{HC} - 12.5\ GAS\ G_{HC}^2$$

$$Pc_{HC} = 677 + 15.0\ GAS\ G_{HC} - 37.5\ GAS\ G_{HC}^2$$

$$GAS\ G_{HC} = \frac{GAS\ G - 0.9672\ y_{N2} - 1.5195\ y_{CO2} - 1.1765\ y_{H2S}}{1 - y_{N2} - y_{CO2} - y_{H2S}}$$

$$Tc = (1 - y_{N2} - y_{CO2} - y_{H2S})\ Tc_{HC} + 227.3\ y_{N2} + 547.6\ y_{CO2} + 672.4\ y_{H2S}$$

$$Pc = (1 - y_{N2} - y_{CO2} - y_{H2S})\ Pc_{HC} + 493.0\ y_{N2} + 1071\ y_{CO2} + 1306\ y_{H2S}$$

Wichert-Aziz Correction:

$$CWA = 120\left[(y_{CO2} + y_{H2S})^{0.9} - (y_{CO2} + y_{H2S})^{1.6}\right] + 15\left(y_{H2S}^{0.5} - y_{H2S}^{4}\right)$$

$$Tc^* = Tc - CWA$$

$$Pc^* = \frac{Pc\,(Tc - CWA)}{Tc + y_{H2S}\,(1 - y_{H2S})\,CWA}$$

$$y_{N2} = \frac{\%N2}{100},\ y_{CO2} = \frac{\%CO2}{100},\ y_{H2S} = \frac{\%H2S}{100}$$

Calculation Subroutines:

There are two calculation subroutines for this program. Both routines return Tc in R to T, Pc in PSI to Z, Tc* in R to Y, Pc* in PSI to X, and CWA in F to LAST X. If %CO2 and %H2S both equal zero, then CWA = 0, Tc = Tc*, and Pc = Pc*.

The first subroutine is called CTcPc (*Calculate Tc and Pc*). The routine expects certain variables (GAS G, %N2, %CO2, %H2S) in the proper registers (see Appendix B), and uses certain scratch registers (see Appendix A).

The second subroutine is called CCWA (*Calculate Wichert-Aziz Correction*). The routine expects certain variables (%CO2, %H2S) in the proper registers (see Appendix B), and uses certain scratch registers (see Appendix A). In addition, Tc in must be in Y and Pc must be in X.

Range of Validity:

$$0 \leqslant \%N2 < 100$$
$$0 \leqslant \%CO2 < 100$$
$$0 \leqslant \%H2S < 100$$
$$0 \leqslant \%N2 + \%CO2 + \%H2S < 100$$

Condensate Fluids:

$$0.56 < \text{GAS G} < 1.30$$

Miscellaneous Gases:

$$0.56 < \text{GAS G} < 1.71$$

Wichert-Aziz Correction:

$$0 \leqslant \%CO2 + \%H2S < 80$$

Remarks:

These correlations for Tc and Pc as a function of GAS G were based on gas mixtures containing negligible amounts of sour gases. The condensate fluids were laboratory-generated gases in equilibrium with crude oil at high pressures (1000-8000 PSI). For these gases with high GAS G, their high gravities come from relatively large quantities of heptanes and heavier compounds. The miscellaneous gases represent surface trap gases and stock-tank vapors whose high gravities result from relatively high propertions of ethane, propane, and butanes.

For these reasons, the condensate fluids correlations should be used in calculations involving gases in equilibrium with crude oil of condensate in petroleum reservoirs, and the miscellaneous gases correlations should be used for surface gases.

It should be kept in mind that these correlations represent only four of an infinite number of possible correlations. There is no unique correlation of pseudocritical properties with gas gravity, and the values calculated are only approximate. If you know the composition of the gas, you will obtain better estimates of Tc and Pc by using the *Gas Properties From Composition* program.

This program, unlike the other programs in the Pac, does not save the output units for you.

Reference:

Standing, M. B., *Volumetric and Phase Behavior of Oil Field Hydrocarbon Systems*, 1977, pp. 26, 122.

User Instructions:

SIZE: 029

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1	Initialize program.		XEQ TcPc	COND? Y/N:‡
2	For condensate fluids, answer "Y". For miscellaneous gases, answer "N".	Y or N	R/S	GAS G=?
3	Key in gas gravity. If you do not know gas gravity, but do know molecular weight, press R/S to get the molecular weight prompt, and then key in molecular weight.	GAS G or	R/S	%N2=?
			R/S	MW=?
		MW	R/S	%N2=?
4	Key in percent nitrogen.	%N2	R/S	%CO2=?
5	Key in percent carbon dioxide.	%CO2	R/S	%H2S=?
6	Key in percent hydrogen sulfide and calculate Tc and Pc. If %CO2 and %H2S both do not equal zero, CWA, Tc*, and Pc* will also be calculated.	%H2S	R/S	Tc=†
			R/S *	Pc=†
			R/S *	CWA=†
			R/S *	Tc*=†
			R/S *	Pc*=†
			R/S *	GAS G=?
7	For a new gas gravity and sour gases, go to step 3. For a new case, go to step 1. Key in only values which change. For values which do not change, ignore the prompts and press R/S.			

* Press R/S if you are not using a printer.

†Press ALPHA to see the units if you are not using a printer.

‡The last character of this prompt will be Y if the prompt is currently true, or N if the prompt is currently false.

Example 1:

Natural gas containing 2 percent nitrogen, 1 percent carbon dioxide, and 7 percent hydrogen sulfide has a gravity of 0.74. Determine the pseudocritical properties of the gas in R and PSI.

Keystrokes (SIZE >= 029)	Display	
XEQ ALPHA TcPc ALPHA	*COND? Y/N:‡*	
N R/S	*GAS G=?*	
.74 R/S	*%N2=?*	
2 R/S	*%CO2=?*	
1 R/S	*%H2S=?*	
7 R/S	*Tc=405.4069*	R†
R/S *	*Pc=714.4590*	PSI†
R/S *	*CWA=14.2174*	F†
R/S *	*Tc*=391.1895*	R†
R/S *	*Pc*=687.8330*	PSI†

Notice that if %CO2 and %H2S both had been equal to zero, CWA, Tc*, and Pc* would not have been calculated. See Example 3.

Now repeat Example 1 of the *Gas Viscosity* program (UG). Take advantage of the fact that most of the variables have already been stored by the TcPc program. Press R/S to skip past the prompts whose values are unchanged from the previous example.

XEQ ALPHA UG ALPHA	*Tc=?*	
R/S	*Pc=?*	
R/S	*GAS G=?*	
R/S	*MW=?*	
R/S	*T=?*	
618 ALPHA R R/S	*P=?*	
125 ALPHA ATM R/S	*UG=0.0164*	CP†

*Press R/S if you are not using a printer.

†Press ALPHA to see the units if you are not using a printer.

‡The last character of this prompt will be Y if the prompt is currently true, or N if the prompt is currently false.

```
Tc Pc

COND: NO
GAS G=0.7400
%N2=2.0000
%CO2=1.0000
%H2S=7.0000

Tc=405.4069 R
Pc=714.4590 PSI
CWA=14.2174 F
Tc*=391.1895 R
Pc*=687.8330 PSI


GAS VIS

T=618.0000 R
P=125.0000 ATM

UG=0.0164 CP
```

For programmers who want to use Pac calculation subroutines in their own programs, the following examples illustrate the use of these routines. If you do not intend to use these subroutines in your programs, you should skip the next examples.

Example 2:

The pseudocritical temperature and pressure for methane are 343 R and 668 PSI. If a sample of methane has the sour gas content of the previous example, what effect do the sour gases have on the pseudocritical properties? Use the second calculation subroutine.

Keystrokes	Display	
343 [ENTER↑] 668		
[XEQ] [ALPHA] CCWA [ALPHA]	***638.5881***	Pc* (PSI)
[R↓]	***328.7826***	Tc* (R)
[R↓]	***668.0000***	Pc (PSI)
[R↓]	***343.0000***	Tc (R)
[LASTx]	***14.2174***	CWA (F)

Example 3:

Compute the pseudocritical properties in R and PSI for this gas if there are no sour gases present.

Keystrokes	Display	
XEQ ALPHA TcPc ALPHA	*COND? Y/N:N*	
R/S	*GAS G=?*	
R/S	*MW=?*	
R/S	*%N2=?*	
0 R/S	*%CO2=?*	
0 R/S	*%H2S=?*	
0 R/S	*Tc=401.6550*	R†
R/S *	*Pc=667.5650*	PSI†
R/S *	*GAS G=?*	

Since %CO2 and %H2S both equal zero, CWA, Tc*, and Pc* were not calculated.

Now use the first calculation subroutine to verify these pseudocritical properties. Since this is a miscellaneous gas like Example 1, and not a condensate fluid, the status set by answering N to the *COND? Y/N* question is still valid for use by the calculation subroutine.

XEQ ALPHA CTcPc ALPHA	*667.5650*	Pc* (PSI)
R↓	*401.6550*	Tc* (R)
R↓	*667.5650*	Pc (PSI)
R↓	*401.6550*	Tc (R)
LASTx	*0.0000*	CWA (F)

Since %CO2 and %H2S both equal zero, CWA = 0, Tc = Tc*, and Pc = Pc*.

```
Tc  Pc

%N2=0.0000
%CO2=0.0000
%H2S=0.0000

Tc=401.6550 R
Pc=667.5650 PSI
```

*Press R/S if you are not using a printer.

†Press ALPHA to see the units if you are not using a printer.

GAS PROPERTIES FROM COMPOSITION

Given the mole percentages of up to 19 different constituents of a natural gas, the PROP program estimates gas gravity (GAS G), pseudocritical temperature (Tc), pseudocritical pressure (Pc), net heating value (NHV), and dry and wet gross heating values (GHVD and GHVW). The calculated values for Tc and Pc are corrected for sour gas content (Tc* and Pc*) with a Wichert-Aziz correction (CWA). The program also estimates, as a function of temperature, the specific heats at constant pressure (CP) and constant volume (CV), and the ratio of specific heats (K).

The allowed gas constituents are sour gases (nitrogen, carbon dioxide, and hydrogen sulfide), methane, ethane, propane, isobutane, n-butane, isopentane, n-pentane, n-hexane, n-heptane, n-octane, n-nonane, n-decane, oxygen, hydrogen, helium, and water vapor. The program automatically stores the mole percentages of the gas constituents, the calculated gas gravity, and the pseudocritical properties for use by other programs in this Pac.

Equations:

Note that in the following equations, the gas property symbols with subscript "i" refer to the properties of each constituent listed in Table 5.

Gas Gravity:

$$\text{GAS G} = \sum_i y_i \, \text{GAS G}_i$$

Pseudocritical Temperature and Pressure:

$$\text{Tc} = \sum_i y_i \, \text{Tc}_i$$

$$\text{Pc} = \sum_i y_i \, \text{Pc}_i$$

Wichert-Aziz Correction:

See the "Equations" section of the *Pseudocritical Temperature and Pressure From Gas Gravity* program.

Heating Values:

$$NHV = \sum_i y_i \, NHV_i$$

$$GHVD = \sum_i y_i \, GHVD_i$$

$$GHVW = 0.9826 \, GHVD$$

Specific Heats and Specific Heat Ratio:

Note that A_i and B_i refer to values in Table 5.

$$CP = \frac{\sum_i y_i (A_i + B_i T')}{28.964 \, GAS \, G}$$

$$CV = CP - \frac{\boldsymbol{R}}{28.964 \, GAS \, G}$$

$$K = \frac{CP}{CV}$$

$$T' = T \text{ in } R$$

$$\boldsymbol{R} = \text{universal gas constant} = \frac{1.987 \, BTU}{LBM \cdot MOL \cdot R}$$

$$y_i = \frac{\%i}{100}$$

Table 5: Properties of Different Natural Gas Constituents

%i	GAS G_i	Tc_i	Pc_i	NHV_i	$GHVD_i$	CP_i	
						A_i	B_i
%N2	0.9672	227.3	493.0	0	0	6.391**	0.09017**
%CO2	1.5195	547.6	1071	0	0	5.87	0.00556
%H2S	1.1765	672.4	1306	588	637	7.16	0.00183
%METH	0.5539	343.04	667.8	909.1	1009.7	5.343	0.006032
%ETH	1.0382	549.76	707.8	1617.8	1768.8	3.782	0.01647
%PROP	1.5225	665.68	616.3	2316.1	2517.4	3.324	0.02662
%IBUT	2.0068	734.65	529.1	3001.1	3252.7	3.857	0.03593
%N-BUT	2.0068	765.32	550.7	3010.4	3262.1	5.104	0.03397
%IPEN	2.4911	828.77	490.4	3698.3	4000.3	4.667	0.04417
%N-PEN	2.4911	845.4	488.6	3707.5	4009.5	6.259	0.04197
%N-HEX	2.9753	913.4	436.9	4403.7	4756.1	6.972	0.05065
%N-HEP	3.4596	972.5	396.8	5100.2	5502.9	8.027	0.05883
%N-OCT	3.9439	1023.89	360.6	5796.7	6249.7	9.13	0.006693
%N-NON	4.4282	1070.35	332	6493.3	6996.6	10.29	0.07496
%N-DEC	4.9125	1111.8	304	7188.6	7742.3	11.4	0.08303
%O2	1.1048	278.6	736.9	0	0	6.545	0.0008859
%H2	0.0696	59.9	188.1	274	324	6.551	0.0005914
%He	0.138	9.5	33.2	0	0	4.97	0
%H2O	0.622	1165.3	3208	0	0	7.587	0.0008195

**For %N2 only, the following is used: A + B ln T'.

Calculation Subroutines:

There are four calculation subroutines for this program. All the subroutines expect the mole percentages of the gas constituents in the proper registers (see Appendix B), and use certain scratch registers (see Appendix A).

The first subroutine, CGASG (*Calculate GAS G*), returns GAS G to X. The second subroutine, CTPC (*Calculate Tc and Pc From Composition*), returns Tc in R to T, Pc in PSI to Z, Tc* in R to Y, Pc* in PSI to X, and CWA in F to LAST X. If %CO2 and %H2S both equal zero, CWA = 0, Tc = Tc*, and Pc = Pc*.

The third subroutine, CHV (*Calculate Heating Values*), returns NHV in BTU/SCF to Z, GHVD in BTU/SCF to Y, and GHVW in BTU/SCF to X. The last subroutine, CCK (*Calculate CP, CV, and K*), expects certain additional variables (GAS G, T) in the proper registers. It returns CP in BTU/LBM*F to Z, CV in BTU/LBM*F to Y, and K to X.

Range of Validity:

$$0 \leqslant \%i \leqslant 100 \text{ for constituent i}$$
$$\%TOT > 0$$

Wichert-Aziz Correction:

See the "Range of Validity" section of the *Pseudocritical Temperature and Pressure From Gas Gravity* program.

CP, CV, K:

$$0 \leqslant T \leqslant 300 \text{ F}$$

Remarks:

Consistent with other programs in this Pac, you may retain the current value of a variable by ignoring the prompt for that variable and pressing R/S. This feature is of particular use when only a few of the possible constituents of a natural gas are present. Set all the percentages to zero by answering Y to the ***CLEAR? Y/N*** prompt, and then ignore the prompts for all components which are not present.

Also, many analyses of a natural gas will not break out the heavier hydrocarbon constituents (C_7-C_{10}). A common practice is to lump all constituents into one mole fraction (e.g., C_{7+}), which could be approximated by the next heavier component (C_8).

In this program, the output units for either the heating values or for the specific heats are saved, depending on the response to the ***SP.HTS? Y/N*** question. If you answer Y, the specific heat units will be saved. If you answer N, the heating value units will be saved.

References:

Kay, W. B., "Density of Hydrocarbon Gases and Vapors," *Industrial and Engineering Chemistry,* Vol. 28, No. 9, pp. 1015-1019.

Gas Processors Suppliers Association, *Engineering Data Book,* 1972, Ninth Edition, Third Revision, pp. 4-1, 16-1, 16-2, 16-3.

Rossini, F. D., *et al., Selected Values of Physical and Thermodynamic Properties of Hydrocarbons and Related Compounds* (API Research Project 44), 1953, p. 653.

Perry, R. H., and Chilton, C. H., *Chemical Engineers' Handbook,* 1973, Fifth Edition, pp. 3-120, 3-121, 3-122.

User Instructions:

SIZE: 045

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1	Initialize program.		XEQ PROP	CLEAR? Y/N:‡
2	If you are starting a new problem, you should set all mole percentages to zero by answering "Y". If you are modifying or correcting a previous problem, you may retain the existing data by answering "N".	Y or N	R/S	%N2=?
	Note: at any point during the entry of the following constituent mole percentages, the remainder of the input list can be skipped by pressing ■ E.			
3	Key in mole percent nitrogen.	%N2	R/S	%CO2=?
4	Carbon Dioxide.	%CO2	R/S	%H2S=?
5	Hydrogen Sulfide.	%H2S	R/S	%METH=?
6	Methane.	%METH	R/S	%ETH=?
7	Ethane.	%ETH	R/S	%PROP=?
8	Propane.	%PROP	R/S	%IBUT=?
9	Isobutane.	%IBUT	R/S	%N-BUT=?
10	N-Butane.	%N-BUT	R/S	%IPEN=?
11	Isopentane.	%IPEN	R/S	%N-PEN=?
12	N-Pentane.	%N-PEN	R/S	%N-HEX=?
13	N-Hexane.	%N-HEX	R/S	%N-HEP=?
14	N-Heptane.	%N-HEP	R/S	%N-OCT=?
15	N-Octane.	%N-OCT	R/S	%N-NON=?
16	N-Nonane.	%N-NON	R/S	%N-DEC=?
17	N-Decane.	%N-DEC	R/S	%O2=?
18	Oxygen.	%O2	R/S	%H2=?
19	Hydrogen.	%H2	R/S	%He=?
20	Helium.	%He	R/S	%H2O=?
21	Key in mole percent water vapor and calculate the total percentage of all constituents.	%H2O	R/S	%TOT=
22	Calculate GAS G.		R/S*	GAS G=

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
23	Calculate Tc and Pc. If %CO2 and %H2S both do not equal zero, CWA, Tc*, and Pc* will also be calculated.		R/S *	Tc=†
			R/S *	Pc=†
			R/S *	CWA=†
			R/S *	Tc*=†
			R/S *	Pc*=†
24	Calculate three heating values.		R/S *	NHV=†
			R/S *	GHVD=†
			R/S *	GHVW=†
			R/S *	SP.HTS? Y/N:‡
25	If you do not want to calculate CP, CV, and K, but instead want to return to the input of the mole percentages, answer "N" and go to step 3. If you do want to calculate CP, CV, and K, answer "Y" and continue.	N or Y	R/S	%N2=?
			R/S	T=?
26	Key in the temperature at which CP, CV, and K will be evaluated, and calculate them.	T	R/S	CP=†
			R/S *	CV=†
			R/S *	K=
			R/S *	T=?
27	For a new temperature, go to step 26. For a new problem, go to step 1. Key in only values which change. For values which do not change, ignore the prompts and press R/S.			

* Press R/S if you are not using a printer.

†Press ALPHA to see the units if you are not using a printer.

‡The last character of this prompt will be Y if the prompt is currently true, or N if the prompt is currently false.

Example 1:

For a mixture of 74 percent methane, 5 percent nitrogen, 3 percent carbon dioxide, 2 percent hydrogen sulfide, 8 percent ethane, 6 percent propane, and 2 percent n-butane, compute the gas properties in Pac default units. Evaluate CP, CV, and K at 50 F.

Keystrokes (SIZE >= 045)	**Display**
XEQ ALPHA PROP ALPHA	*CLEAR? Y/N:‡*
Y R/S	*%N2=?*
5 R/S	*%CO2=?*
3 R/S	*%H2S=?*
2 R/S	*%METH=?*
74 R/S	*%ETH=?*
8 R/S	*%PROP=?*
6 R/S	*%IBUT=?*
R/S	*%N-BUT=?*
2 R/S	*%IPEN=?*
R/S	*%N-PEN=?*
R/S	*%N-HEX=?*

After inputting all of the mole percentages for the gas you are analyzing, there is an option that allows you to skip the remainder of the input list. This option is invoked by pressing ■ E (assuming there is nothing assigned to that key). This option can be used whenever you are entering a mole percentage for this program.

■ E	*%TOT=100.0000*

You should note that when using this "skip" option while keying in a new value, it doesn't matter whether you press ■ E before or after pressing R/S. So when you keyed in the mole percentage for n-butane, you could have pressed either 2 R/S ■ E or 2 ■ E. Either keystroke sequence would have entered the 2 and skipped the remainder of the input list.

You should also note that %TOT may be larger or smaller than 100, depending on the mole percentages provided by your gas analysis. You may wish to adjust the mole percentages slightly to produce a total of 100%.

Keystrokes	Display	
R/S *	*GAS G=0.7419*	
R/S *	*Tc=394.3186*	R†
R/S *	*Pc=681.6880*	PSI†
R/S *	*CWA=9.2227*	F†
R/S *	*Tc*=385.0959*	R†
R/S *	*Pc*=665.4390*	PSI†

Notice that if %CO2 and %H2S both had been equal to zero, CWA, Tc*, and Pc* would not have been calculated.

R/S *	*NHV=1013.0920*	BTU/SCF†
R/S *	*GHVD=1117.7080*	BTU/SCF†
R/S *	*GHVW=1098.2599*	BTU/SCF†
R/S *	*SP.HTS? Y/N:‡*	
Y R/S	*T=?*	

Key in the temperature at which CP, CV, and K will be evaluated.

50 R/S	*CP=0.4391*	BTU/LBM*F†
R/S *	*CV=0.3466*	BTU/LBM*F†
R/S *	*K=1.2668*	

*Press R/S if you are not using a printer.

†Press ALPHA to see the units if you are not using a printer.

‡The last character of this prompt will be Y if the prompt is currently true, or N if the prompt is currently false.

```
GAS PROP

CLEAR: YES
%N2=5.0000
%CO2=3.0000
%H2S=2.0000
%METH=74.0000
%ETH=8.0000
%PROP=6.0000
%N-BUT=2.0000

%TOT=100.0000
GAS G=0.7419
Tc=394.3186 R
Pc=681.6880 PSI
CWA=9.2227 F
Tc*=385.0959 R
Pc*=665.4390 PSI
NHV=1013.0920 BTU/SCF
GHVD=1117.7080 BTU/SCF
GHVW=1098.2599 BTU/SCF

SP.HTS: YES
T=50.0000 F
CP=0.4391 BTU/LBM*F
CV=0.3466 BTU/LBM*F
K=1.2668
```

For programmers who want to use Pac input and calculation subroutines in their own programs, the following example illustrates the use of these routines. If you do not intend to use these subroutines in your programs, you should skip the next example.

Example 2:

Suppose the gas in question has the same total sour gas content, but in different proportions: 0 percent nitrogen, 1 percent carbon dioxide, and 9 percent hydrogen sulfide. Using the calculation subroutines, recalculate the gas gravity, pseudocritical properties, heating values, specific heats, and specific heat ratio, all in Pac default units. Use the sour gas input routine (SOUR) to input and store the new sour gas percentages.

Keystrokes	Display	
XEQ ALPHA SOUR ALPHA	*%N2=?*	
0 R/S	*%CO2=?*	
1 R/S	*%H2S=?*	
9 R/S	*9.0000*	%H2S
XEQ ALPHA CGASG ALPHA	*0.7455*	GAS G
XEQ ALPHA CTPC ALPHA	*695.9962*	Pc* (PSI)
R↓	*402.4777*	Tc* (R)
R↓	*727.0380*	Pc (PSI)
R↓	*419.0696*	Tc (R)
■ LAST x	*16.5919*	CWA (F)
XEQ ALPHA CHV ALPHA	*1142.0740*	GHVW (BTU/SCF)
R↓	*1162.2980*	GHVD (BTU/SCF)
R↓	*1054.2520*	NHV (BTU/SCF)

Now save the calculated gas gravity for use by the CCK subroutine. (Remember that a calculation subroutine returns values to the stack only. The CGASG subroutine calculated GAS G, but did not store it.)

XEQ ALPHA GASG ALPHA	*GAS G=?*	
.7455 R/S	*0.7455*	GAS G
XEQ ALPHA CCK ALPHA	*1.2652*	K
R↓	*0.3470*	CV (BTU/LBM*F)
R↓	*0.4390*	CP (BTU/LBM*F)

```
%N2=0.0000
%CO2=1.0000
%H2S=9.0000
GAS G=0.7455
```

OIL ISOTHERMAL COMPRESSIBILITY

The CO program estimates the oil isothermal coefficient of compressibility above and below the bubble point (CO and COb).

Equations:

Above Bubble Point:

$$CO = \frac{-1433.0 + 5.0\,RSI + 17.2\,T - 1180.0\,GAS\,GS + 12.61\,OIL\,G}{10^5\,P}$$

$$GAS\,GS = GAS\,G\left[1 + 5.912\,(10^{-5})\,OIL\,G\,(SEP\,T)\log\frac{SEP\,P}{114.7}\right]$$

Below Bubble Point:

$$COb = \frac{-1}{BOb}\frac{\partial BOb}{\partial P} + \frac{BG'}{BOb}\frac{\partial RSb}{\partial P}$$

$$BG' = BG \text{ in BBL/SCF}$$

Calculation Subroutines:

There are three calculation subroutines for this program. The first is called CCOb (*Calculate CO Below Bubble Point*). The routine expects certain variables (STD T, STD P, SEP T, SEP P, OIL G, GAS G, T) in the proper registers (see Appendix B), and uses certain scratch registers (see Appendix A). In addition, TR must be in Z, PR must be in Y, and P must be in X. Upon return, COb in 1/PSI will be in X, and RSb in SCF/BBL will be in Y.

The second subroutine is called CCO (*Calculate CO Above Bubble Point*). The routine expects certain variables (SEP T, SEP P, OIL G, GAS G, T, RSI, P) in the proper registers (see Appendix B). Upon return, CO in 1/PSI will be in X.

The third subroutine is called CGS (*Calculate Gas Gravity Corrected for Separator Conditions*). The routine expects certain variables (SEP T, SEP P, OIL G, GAS G) in the proper registers (see Appendix B). Upon return, GAS GS will be in X.

Range of Validity:

$$76 < SEP\,T < 150\ F$$
$$30 < SEP\,P < 535\ PSI$$
$$15.3 < OIL\,G < 59.5\ API$$

Above Bubble Point:

$$0.511 < \text{GAS G} < 1.351$$
$$111 < \text{P} < 9485 \text{ PSI}$$

Below Bubble Point:

For $15.3 < \text{OIL G} \leqslant 30$ API:

$$0.511 < \text{GAS G} < 1.351$$
$$14.7 < \text{P} < 4542 \text{ PSI}$$

For $30.6 < \text{OIL G} < 59.5$ API:

$$0.530 < \text{GAS G} < 1.259$$
$$14.7 < \text{P} < 6025 \text{ PSI}$$

Below the bubble point, see also the "Range of Validity" section of the *Z Factor* program.

Remarks:

The correlations used for estimating RSb and BOb do not take into account the discontinuity at the bubble point. Consequently, the derivatives ∂RSb/∂P, ∂BOb/∂RSb, and ∂BOb/∂P do not accurately reflect the behavior of RSb and BOb approaching the bubble point. Because of this, COb may be larger than expected at pressures slightly below the bubble point.

If you know that the pressures of interest are greater than the bubble point, you do not have to input values for Tc, Pc, STD T, or STD P. Just R/S past those prompts.

This program is one of several that correct the gas gravity for separator temperature and pressure (SEP T and SEP P) by using the CGS subroutine. If you do not know the separator conditions, you can eliminate the effect of the correction by using 0 F for SEP T (with any positive, nonzero pressure) or 114.7 PSI for SEP P.

References:

Ramey, H. J., "Rapid Methods for Estimating Reservoir Compressibilities," *Journal of Petroleum Technology,* April, 1964, pp. 447-454.

Vasquez, M., and Beggs, H. D., "Correlations for Fluid Physical Property Predictions," *Journal of Petroleum Technology,* June, 1980, pp. 968-970.

User Instructions:

				SIZE: 026
STEP	**INSTRUCTIONS**	**INPUT**	**FUNCTION**	**DISPLAY**
1	Initialize program.		XEQ CO	Tc=?
2	Key in pseudocritical temperature.	Tc	R/S	Pc=?
3	Key in pseudocritical pressure.	Pc	R/S	STD T=?
4	Key in temperature at standard conditions. If the previously stored STD T is 0 F, it will be replaced by 60 F before this prompt appears.	STD T	R/S	STD P=?

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
5	Key in pressure at standard conditions. If the previously stored STD P is 0 PSI, it will be replaced by 14.65 PSI before this prompt appears.	STD P	R/S	SEP T=?
6	Key in separator temperature.	SEP T	R/S	SEP P=?
7	Key in separator pressure.	SEP P	R/S	OIL G=?
8	Key in oil gravity.	OIL G	R/S	GAS G=?
9	Key in gas gravity and calculate GAS GS. If you do not know gas gravity, but do know molecular weight, press R/S to get the molecular weight prompt, and then key in molecular weight.	GAS G	R/S R/S*	GAS GS= T=?
		or		
		MW	R/S R/S R/S*	MW=? GAS GS= T=?
10	Key in temperature.	T	R/S	RSI=?
11	Key in initial gas-oil ratio and calculate PBP.	RSI	R/S R/S*	PBP=† P=?
12	Key in pressure. If the pressure is above the bubble point, CO will be calculated. If the pressure is below the bubble point, RSb for that pressure will be calculated, followed by COb.	P	R/S R/S*	CO=† P=?
		P	R/S R/S* R/S*	RSb=† COb=† P=?
13	For a new pressure, go to step 12. For a new case, go to step 1. Key in only values which change. For values which do not change, ignore the prompts and press R/S. * Press R/S if you are not using a printer. †Press ALPHA to see the units if you are not using a printer.			

Example 1:

A reservoir oil at a temperature of 68 C has a gravity of 39.8 API and an initial gas-oil ratio of 82 SCM/M3. The dissolved gas gravity is 0.83, Tc is 240 K, and Pc is 4580 KPA. Separator conditions are 38 C and 860 KPA, and the standard conditions for the gas are 15.6 C and 1 ATM. What is the isothermal compressibility for the oil in 1/KPA at 70 and 140 ATM?

There are two ways you can solve this problem. The first is like the other examples in the Pac, where the default units are English, and you change the units as necessary. Also, you would turn on the output

portion of the Unit Management System to allow you to change the output units to 1/KPA instead of the default 1/PSI.

The second way is to use the SI option of the Unit Management System. With this option, all the input and output prompts will use SI units automatically instead of English units. Turn on the SI option (set flag 09) for this example.

Note that you still have to enter those units which do not match either the English or the SI default units. If you do not remember what the default units are when responding to an input prompt, you can always check by pressing [ALPHA] to see the units. When responding to an output prompt, the default units will be displayed as part of the prompt (assuming flag 10 is set). See Appendix C for a list of the default English and SI units for each variable used in the Pac.

Keystrokes (SIZE >= 026)	**Display**
[SF] 09 [ENG] 4	
[XEQ] [ALPHA] CO [ALPHA]	*Tc=?*
240 [ALPHA]	*K*

Notice that with flag 09 set, the default units in the ALPHA register are now SI units. As you follow the remainder of the example, you will see that both the input and output default units have changed.

[R/S]	*Pc=?*	
4580 [ALPHA]	*KPA*	
[R/S]	*STD T=?*	
15.6 [ALPHA]	*C*	
[R/S]	*STD P=?*	
1 [ALPHA] ATM [R/S]	*SEP T=?*	
38 [R/S]	*SEP P=?*	(absolute P)
860 [R/S]	*OIL G=?*	
39.8 [ALPHA]	*KG/M3*	
API [R/S]	*GAS G=?*	
.83 [R/S]	*GAS GS=837.14E-3*	
[R/S] *	*T=?*	
68 [R/S]	*RSI=?*	
82 [ALPHA]	*SCM/M3*	
[R/S]	*PBP=11.332E3*	KPA†
[R/S] *	*P=?*	
70 [ALPHA] ATM [R/S]	*RSb=47.019E0*	SCM/M3†
[R/S] *	*COb=75.187E-6*	1/KPA†
[R/S] *	*P=?*	
140 [ALPHA] ATM [R/S]	*CO=2.1453E-6*	1/KPA†

Now determine the compressibility at the 140 ATM pressure in 1/PSI. Turn on the output portion of the Unit Management System (set flag 10) to allow you to change the output units.

R/S *	*P=?*	
SF 10 R/S	*CO, 1/KPA?*	
1/PSI R/S	*CO=14.791E-6*	1/PSI†

```
OIL ISO CMP

Tc=240.00E0 K
Pc=4.5800E3 KPA
STD T=15.600E0 C
STD P=1.0000E0 ATM
SEP T=38.000E0 C
SEP P=860.00E0 KPA
OIL G=39.800E0 API
GAS G=830.00E-3
GAS GS=837.14E-3
T=68.000E0 C
RSI=82.000E0 SCM/M3
PBP=11.332E3 KPA
P=70.000E0 ATM
RSb=47.019E0 SCM/M3

COb=75.187E-6 1/KPA

P=140.00E0 ATM
CO=2.1453E-6 1/KPA

CO=14.791E-6 1/PSI
```

Note that to determine the compressibility in 1/PSI, you could have done the following:

2.1453 EEX CHS 6	*2.1453 -6_*	CO (1/KPA)
ALPHA 1/KPA-1/PSI ALPHA	*2.1453 -06*	
XEQ ALPHA CON ALPHA	*14.791 -06*	CO (1/PSI)

You could have done the same kind of conversion using CON to determine the equivalent of each input variable in English units.

*Press R/S if you are not using a printer.

†Press ALPHA to see the units if you are not using a printer.

For programmers who want to use Pac input and calculation subroutines in their own programs, the following examples illustrate the use of these routines. If you do not intend to use these subroutines in your programs, you should skip the next examples.

Example 2:

Assuming you have just run the last example, use the calculation subroutines to compute the isothermal compressibility in 1/PSI at a gas-oil ratio of 72 SCM/M3 and the pressures in the last example. Use the gas-oil ratio input routine (RSI) to input and store the new RSI. Then use the CPBP calculation subroutine to determine whether the pressures from the example are above or below the bubble point.

Keystrokes	Display	
XEQ ALPHA RSI ALPHA	*RSI=?*	
72 R/S	*405.03 00*	RSI (SCF/BBL)
XEQ ALPHA CPBP ALPHA	*1.4730 03*	PBP (PSI)
ALPHA PSI-ATM ALPHA	*1.4730 03*	
XEQ ALPHA CON ALPHA	*100.23 00*	PBP (ATM)

(Remember that all input and calculation subroutines return values in Pac English default units.) Since the last pressure used in Example 1 is greater than the bubble point just calculated, use CCO to calculate the compressibility.

XEQ ALPHA CCO ALPHA	*13.424 -06*	CO (1/PSI)

Now use the P input routine to input and store the 70 ATM pressure from Example 1. Since this pressure is less than the bubble point just calculated, use CCOb to calculate the compressibility.

XEQ ALPHA P ALPHA	*P=?*	
70 ALPHA ATM R/S	*1.0287 03*	P (PSI)

The CCOb routine requires TR in Z, PR in Y, and P in X, which are the stack contents after executing P.

XEQ ALPHA CCOb ALPHA	*518.40 -06*	COb (1/PSI)
R↓	*264.50 00*	RSb (SCF/BBL)
CF 09 CF 10 FIX 4	*264.5003*	

```
RSI=72.000E0 SCM/M3
P=70.000E0 ATM
```

Example 3:

Using the CGS calculation subroutine, correct the gas gravity to account for separator conditions.

Keystrokes	Display	
XEQ ALPHA CGS ALPHA	*0.8371*	GAS GS

Notes

OIL FORMATION VOLUME FACTOR

The BO program estimates the oil formation volume factor at the bubble point (BOBP), above the bubble point (BO), and below the bubble point (BOb). For use in the Pac, BOBP, BO and BOb are defined as the ratio of the liquid volume at reservoir conditions to the liquid volume at stock tank conditions.

Equations:

Above Bubble Point:

$$BO = BOBP \exp [CO (PBP - P)]$$

Below Bubble Point:

$$BOb = 1 + A (T - 60) (OIL\ G/GAS\ GS) + [B + C (T - 60) (OIL\ G/GAS\ GS)] RSb$$

$$\partial BOb/\partial RSb = B + C (T - 60) (OIL\ G/GAS\ GS)$$

At Bubble Point:

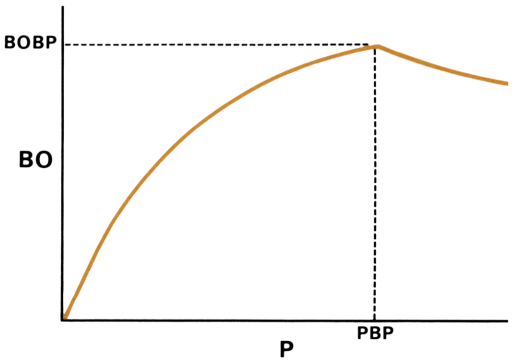
$$BOBP = BOb \text{ at } P = PBP$$

For OIL G $\leqslant$ 30 API:

$A = 1.751 \times 10^{-5}$
$B = 4.677 \times 10^{-4}$
$C = -1.811 \times 10^{-8}$

For OIL G $>$ 30 API:

$A = 1.100 \times 10^{-5}$
$B = 4.670 \times 10^{-4}$
$C = 1.337 \times 10^{-9}$



Calculation Subroutines:

There are two calculation subroutines for this program. The first is called CBOb (*Calculate BO Below Bubble Point*). The routine expects certain variables (SEP T, SEP P, OIL G, GAS G, T) in the proper registers (see Appendix B), and uses certain scratch registers (see Appendix A). In addition, a pressure must be in X. Upon return, BOb or BOBP will be in X, depending on whether P or PBP, respectively, was in X when the routine was called. The partial derivative of BOb with respect to RSb at constant T ($\partial BOb/\partial RSb$) will be in Y, the partial derivative of RSb with respect to P at constant T ($\partial RSb/\partial P$) will be in Z, and RSb in SCF/BBL will be in T.

The second subroutine is called CBO (*Calculate BO Above Bubble Point*). The routine expects certain variables (SEP T, SEP P, OIL G, GAS G, T, RSI, PBP, P) in the proper registers (see Appendix B). In addition, BOBP must be in X. Upon return, BO will be in X.

Range of Validity:

$$76 < \text{SEP T} < 150 \text{ F}$$
$$30 < \text{SEP P} < 535 \text{ PSI}$$
$$15.3 < \text{OIL G} < 59.5 \text{ API}$$

Above Bubble Point:

$$0.511 < \text{GAS G} < 1.351$$
$$111 < \text{P} < 9485 \text{ PSI}$$

At or Below Bubble Point:

For $15.3 < \text{OIL G} \leq 30$ API:	For $30.6 < \text{OIL G} < 59.5$ API
$0.511 < \text{GAS G} < 1.351$	$0.530 < \text{GAS G} < 1.259$
$14.7 < \text{P} < 4542$ PSI	$14.7 < \text{P} < 6025$ PSI

At or below the bubble point, see also the "Range of Validity" section of the *Z Factor* program.

Remarks:

The correlations used for estimating BOb do not take into account the discontinuity at the bubble point. Consequently, the derivative $\partial BOb/\partial RSb$ does not accurately reflect the behavior of BOb approaching the bubble point. Because of this, BOb may be larger than expected at pressures slightly below the bubble point.

References:

Ramey, H. J., "Rapid Methods for Estimating Reservoir Compressibilities," *Journal of Petroleum Technology,* April, 1964, pp. 447-454.

Vasquez, M., and Beggs, H. D., "Correlations for Fluid Physical Property Predictions," *Journal of Petroleum Technology,* June, 1980, pp. 968-970.

User Instructions:

SIZE: 026

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1	Initialize program.		XEQ BO	SEP T=?
2	Key in separator temperature.	SEP T	R/S	SEP P=?
3	Key in separator pressure.	SEP P	R/S	OIL G=?
4	Key in oil gravity.	OIL G	R/S	GAS G=?
5	Key in gas gravity and calculate GAS GS. If you do not know gas gravity, but do know molecular weight, press R/S to get the molecular weight prompt, and then key in molecular weight.	GAS G	R/S R/S*	GAS GS= T=?
		or		
		MW	R/S R/S R/S*	MW=? GAS GS= T=?
6	Key in temperature.	T	R/S	RSI=?
7	Key in initial gas-oil ratio and calculate PBP and BOBP.	RSI	R/S R/S* R/S*	PBP=† BOBP= P=?
8	Key in pressure. If the pressure is above the bubble point, BO will be calculated. If the pressure is below the bubble point, RSb for that pressure will be calculated, followed by BOb.	P	R/S R/S*	BO= P=?
		P	R/S R/S* R/S*	RSb=† BOb= P=?
9	For a new pressure, go to step 8. For a new case, go to step 1. Key in only values which change. For values which do not change, ignore the prompts and press R/S.			
	* Press R/S if you are not using a printer.			
	†Press ALPHA to see the units if you are not using a printer.			

Example 1:

A reservoir oil has a gravity of 39.8 API and an initial gas-oil ratio of 460 SCF/BBL at a temperature of 155 F. The dissolved gas has a gravity of 0.83, measured at separator conditions of 100 F and 125 PSI. What is the formation volume factor for the oil at 1000 and 2000 PSI? (These are the same values used in Example 1 of the *Two-Phase Formation Volume Factor* program. If you just ran that example, the only values you will need to enter are the pressures. Just [R/S] past the other prompts.)

Keystrokes (SIZE >= 026)	Display	
[XEQ] [ALPHA] BO [ALPHA]	*SEP T=?*	
100 [R/S]	*SEP P=?*	(absolute P)
125 [R/S]	*OIL G=?*	
39.8 [R/S]	*GAS G=?*	
.83 [R/S]	*GAS GS=0.8373*	
[R/S] *	*T=?*	
155 [R/S]	*RSI=?*	
460 [R/S]	*PBP=1641.5439*	PSI†
[R/S] *	*BOBP=1.2673*	
[R/S] *	*P=?*	
1000 [R/S]	*RSb=255.4189*	SCF/BBL†
[R/S] *	*BOb=1.1705*	
[R/S] *	*P=?*	
2000 [R/S]	*BO=1.2604*	

*Press [R/S] if you are not using a printer.

†Press [ALPHA] to see the units if you are not using a printer.

```
OIL VOL FACT

SEP T=100.0000 F
SEP P=125.0000 PSI
OIL G=39.8000 API
GAS G=0.8300
GAS GS=0.8373
T=155.0000 F
RSI=460.0000 SCF/BBL
PBP=1641.5439 PSI

BOBP=1.2673

P=1000.0000 PSI
RSb=255.4189 SCF/BBL
BOb=1.1705

P=2000.0000 PSI
BO=1.2604
```

For programmers who want to use Pac input and calculation subroutines in their own programs, the following examples illustrate the use of these routines. If you do not intend to use these subroutines in your programs, you should skip the next examples.

Example 2:

Assuming you have just run the last example, use the first calculation subroutine to recompute the formation volume factor and its associated partial derivatives. Use the pressure input routine (P) to input and store the first pressure.

Keystrokes	Display	
XEQ ALPHA P ALPHA	*P=?*	
1000 R/S	*1000.0000*	P (PSI)

The CBOb routine requires P in X.

XEQ ALPHA CBOb ALPHA	*1.1705*	BOb
R↓	*0.0005*	∂BOb/∂RSb
R↓	*0.3032*	∂RSb/∂P
R↓	*255.4189*	RSb (SCF/BBL)

```
P=1000.0000 PSI
```

Example 3:

If the oil has an initial gas-oil ratio of 200 SCF/BBL, what is the formation volume factor? Press [R/S] to skip past the prompts whose values are unchanged from the previous examples. If you want to verify what the existing value is for a particular variable, press [←] when the prompt appears.

Keystrokes	Display	
[XEQ] [ALPHA] BO [ALPHA]	*SEP T=?*	
[R/S]	*SEP P=?*	
[R/S]	*OIL G=?*	
[R/S]	*GAS G=?*	
[R/S]	*MW=?*	
[R/S]	*GAS GS=0.8373*	
[R/S] *	*T=?*	
[R/S]	*RSI=?*	
200 [R/S]	*PBP=813.7880*	PSI†
[R/S] *	*BOBP=1.1443*	
[R/S] *	*P=?*	
[R/S]	*BO=1.1406*	

Now verify the result by using the appropriate calculation subroutine. Since the pressure used in the last example is greater than the bubble point just calculated, use CBO. The routine requires BOBP in X.

1.1443 [XEQ] [ALPHA] CBO [ALPHA]	*1.1406*	BO

```
OIL VOL FACT

GAS GS=0.8373
RSI=200.0000 SCF/BBL
PBP=813.7880 PSI

BOBP=1.1443

BO=1.1406
```

*Press [R/S] if you are not using a printer.

†Press [ALPHA] to see the units if you are not using a printer.

OIL VISCOSITY

The UO program estimates dead oil viscosity (UOd) and live oil viscosity at the bubble point (UOBP). Also, the live oil viscosity above (UO) and below (UOb) the bubble point as a function of pressure is estimated.

Equations:

Dead Oil:

$$UOd = 10^{A} - 1$$

$$A = B\,T^{-1.163}$$

$$B = 10^{C}$$

$$C = 3.0324 - 0.02023\ OIL\ G$$

Live Oil:

Above Bubble Point:

$$UO = UOBP\,(P/PBP^{A})$$

$$A = 2.6\,P^{1.187}\exp\left[-8.98\,(10^{-5})\,P - 11.513\right]$$

Below Bubble Point:

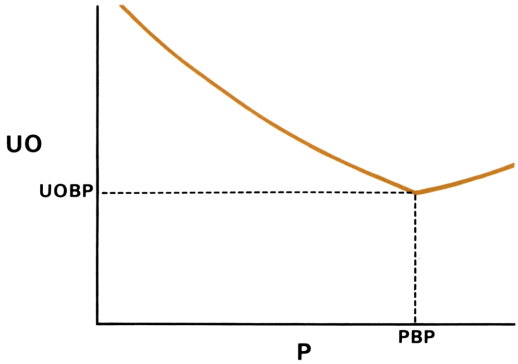
$$UOb = A\,UOd^{B}$$

$$A = 10.715\,(RSb + 100)^{-0.515}$$

$$B = 5.44\,(RSb + 150)^{-0.338}$$

At Bubble Point:

$$UOBP = UOb \text{ at } RSb = RSI$$



Calculation Subroutines:

There are three calculation subroutines for this program. The first is called CUOd (*Calculate UO for Dead Oil*). The routine expects certain variables (OIL G, T) in the proper registers (see Appendix B). Upon return, UOd in CP will be in X.

The second subroutine is called CUOb (*Calculate UO for Live Oil Below Bubble Point*). The routine expects a gas-oil ratio in Y and UOd in X. Upon return, UOb or UOBP in CP will be in X, depending on whether RSb or RSI, respectively, was in Y when the routine was called.

The third subroutine is called CUO (*Calculate UO for Live Oil Above Bubble Point*). The routine expects certain variables (PBP, P) in the proper registers (see Appendix B). In addition, UOBP must be in X. Upon return, UO in CP will be in X.

Range of Validity:

Dead Oil:

$$16 < \text{OIL G} < 58 \text{ API}$$
$$70 < \text{T} < 295 \text{ F}$$

Live Oil:

$$76 < \text{SEP T} < 150 \text{ F}$$
$$30 < \text{SEP P} < 535 \text{ PSI}$$

Above Bubble Point:

$$15.3 < \text{OIL G} < 59.5 \text{ API}$$
$$0.511 < \text{GAS G} < 1.351$$
$$111 < \text{P} < 9485 \text{ PSI}$$

At or Below Bubble Point:

$$20 < \text{RSb} < 2070 \text{ SCF/BBL}$$
$$14.7 < \text{P} < 5265 \text{ PSI}$$
$$70 < \text{T} < 295 \text{ F}$$
$$16 < \text{OIL G} < 58 \text{ API}$$

References:

Beggs, H. D., Robinson, J. R., "Estimating the Viscosity of Crude Oil Systems," *JPT Forum,* September, 1975, pp. 1140-1141.

Vasquez, M. and Beggs, H. D., "Correlations for Fluid Physical Property Predictions," *Journal of Petroleum Technology,* June, 1980, pp. 968-970.

User Instructions:

				SIZE: 026
STEP	**INSTRUCTIONS**	**INPUT**	**FUNCTION**	**DISPLAY**
1	Initialize program.		XEQ UO	SEP T=?
2	Key in separator temperature.	SEP T	R/S	SEP P=?
3	Key in separator pressure.	SEP P	R/S	OIL G=?
4	Key in oil gravity.	OIL G	R/S	GAS G=?
5	Key in gas gravity and calculate GAS GS. If you do not know gas gravity, but do know molecular weight, press R/S to get the molecular weight prompt, and then key in molecular weight.	GAS G or MW	R/S R/S* R/S R/S R/S*	GAS GS= T=? MW=? GAS GS= T=?
6	Key in temperature and calculate UOd.	T	R/S R/S*	UOd=† RSI=?
7	Key in initial gas-oil ratio and calculate PBP and UOBP.	RSI	R/S R/S* R/S*	PBP=† UOBP=† P=?

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
8	Key in pressure. If the pressure is above the bubble point, UO will be calculated. If the pressure is below the bubble point, RSb for that pressure will be calculated, followed by UOb.	P	R/S R/S*	UO=† P=?
		P	R/S R/S* R/S*	RSb=† UOb=† P=?
9	For a new pressure, go to step 8. For a new case, go to step 1. Key in only values which change. For values which do not change, ignore the prompts and press R/S.			
	*Press R/S if you are not using a printer.			
	†Press ALPHA to see the units if you are not using a printer.			

Example 1:

A reservoir oil has a gravity of 38 API and an initial gas-oil ratio of 450 SCF/BBL at a temperature of 150 F. The dissolved gas has a gravity of 0.64, measured at separator conditions of 100 F and 125 PSI. What is the viscosity of the oil at 1500 and 2300 PSI?

Keystrokes (SIZE >= 026)	Display	
XEQ ALPHA UO ALPHA	*SEP T=?*	
100 R/S	*SEP P=?*	(absolute P)
125 R/S	*OIL G=?*	
38 R/S	*GAS G=?*	
.64 R/S	*GAS GS=0.6454*	
R/S *	*T=?*	
150 R/S	*UOd=2.4721*	CP†
R/S *	*RSI=?*	
450 R/S	*PBP=2107.0188*	PSI†
R/S *	*UOBP=0.7324*	CP†
R/S *	*P=?*	
1500 R/S	*RSb=300.6342*	SCF/BBL†
R/S *	*UOb=0.9133*	CP†
R/S *	*P=?*	
2300 R/S	*UO=0.7458*	CP†

*Press R/S if you are not using a printer.

†Press ALPHA to see the units if you are not using a printer.

```
OIL VIS

SEP T=100.0000 F
SEP P=125.0000 PSI
OIL G=38.0000 API
GAS G=0.6400
GAS GS=0.6454
T=150.0000 F

UOd=2.4721 CP

RSI=450.0000 SCF/BBL
PBP=2107.0188 PSI
UOBP=0.7324 CP

P=1500.0000 PSI
RSb=300.6342 SCF/BBL
UOb=0.9133 CP

P=2300.0000 PSI
UO=0.7458 CP
```

For programmers who want to use Pac input and calculation subroutines in their own programs, the following examples illustrate the use of these routines. If you do not intend to use these subroutines in your programs, you should skip the next examples.

Example 2:

Assuming you have just run the last example, use the CUO calculation subroutine to verify the final result. The routine requires UOBP in X.

Keystrokes	Display	
.7324 [XEQ] [ALPHA] CUO [ALPHA]	***0.7458***	UO (CP)

Example 3:

Calculate the dead oil viscosity of the oil in Example 1 if it has a specific gravity of 0.876. Also compute the bubble point viscosity for that oil. Use the oil gravity input routine (OILG) to input and store the new oil gravity.

Keystrokes	Display	
XEQ ALPHA OILG ALPHA	*OIL G=?*	
.876 ALPHA SPGR R/S	*30.0297*	OIL G (API)
R↓	*SPGR*	Units entered
R↓	*(blank)*	by user

Notice that the value returned to the stack is in Pac English default units, and that the units entered by the user are in Y and Z. Since the default units for each value returned in the stack are known, the ALPHA register will not contain the units for those values.

XEQ ALPHA CUOd ALPHA	*5.0760*	UOd (CP)

(Remember that all input and calculation subroutines return values in Pac English default units.) Now calculate the viscosity at the bubble point. The CUOb subroutine expects RSI in Y and UOd in X.

450	*450_*	RSI (SCF/BBL)
$x \gtrless y$ XEQ ALPHA CUOb ALPHA	*1.1491*	UOb (CP)

OIL G=0.8760 SPGR

GAS-OIL RATIO

Given separator conditions, oil and gas gravity, temperature, initial gas-oil ratio, and pressure, the RS program estimates the gas-oil ratio below the bubble point (RSb).

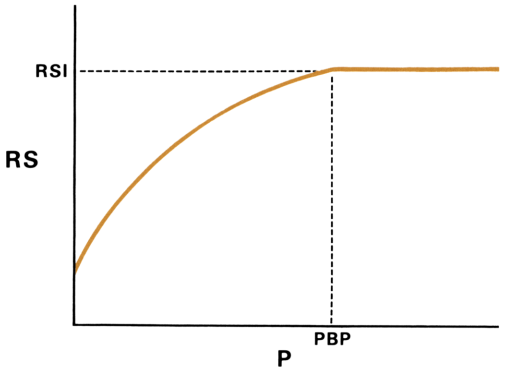
Equations:

$$RSb = A\ GAS\ GS\ P^{B} \exp(C\ OIL\ G/T')$$

$$\partial RSb/\partial P = \frac{B\ RSb}{P}$$

$$T' = T \text{ in } R$$

For OIL G $\leq$ 30 API:	For OIL G $>$ 30 API:
A = 0.0362	A = 0.0178
B = 1.0937	B = 1.1870
C = 25.7240	C = 23.9310



Calculation Subroutine:

The calculation subroutine for the gas-oil ratio program is called CRSb (*Calculate RS Below Bubble Point*). The routine expects certain variables (SEP T, SEP P, OIL G, GAS G, T, P) in the proper registers (see Appendix B). In addition, P must be in X. Upon return, RSb in SCF/BBL will be in X, and the partial derivative of RSb with respect to P at constant T ($\partial RSb/\partial P$) will be in Y. There is no calculation subroutine to calculate RS above the bubble point, since above the bubble point, RS = RSI.

Range of Validity:

$$76 < \text{SEP T} < 150 \text{ F}$$
$$30 < \text{SEP P} < 535 \text{ PSI}$$

For $15.3 < \text{OIL G} \leq 30$ API:

$0.511 < \text{GAS G} < 1.351$
$14.7 < \text{P} < 4542$ PSI

For $30.6 < \text{OIL G} < 59.5$ API:

$0.530 < \text{GAS G} < 1.259$
$14.7 < \text{P} < 6025$ PSI

Reference:

Vasquez, M., and Beggs, H. D., "Correlations for Fluid Physical Property Predictions," *Journal of Petroleum Technology*, June, 1980, pp. 968-970.

User Instructions:

SIZE: 026

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1	Initialize program.		XEQ RS	SEP T=?
2	Key in separator temperature.	SEP T	R/S	SEP P=?
3	Key in separator pressure.	SEP P	R/S	OIL G=?
4	Key in oil gravity.	OIL G	R/S	GAS G=?
5	Key in gas gravity and calculate GAS GS. If you do not know gas gravity, but do know molecular weight, press R/S to get the molecular weight prompt, and then key in molecular weight.	GAS G	R/S R/S*	GAS GS= T=?
		or		
		MW	R/S R/S R/S*	MW=? GAS GS= T=?
6	Key in temperature.	T	R/S	RSI=?
7	Key in initial gas-oil ratio and calculate PBP.	RSI	R/S R/S*	PBP=† P=?
8	Key in pressure. If the pressure is above the bubble point, RS will be displayed. If the pressure is below the bubble point, RSb for that pressure will be calculated.	P	R/S R/S*	RS=† P=?
		P	R/S R/S*	RSb=† P=?
9	For a new pressure, go to step 8. For a new case, go to step 1. Key in only values which change. For values which do not change, ignore the prompts and press R/S.			
	* Press R/S if you are not using a printer.			
	†Press ALPHA to see the units if you are not using a printer.			

Example 1:

The initial gas-oil ratio of a 30 API oil is 350 SCF/BBL. The oil temperature is 200 F, and the gravity of the dissolved gas is 0.75. Separator conditions are 100 F and 125 PSI. What is the gas-oil ratio at 1200 and 2100 PSI?

Keystrokes (SIZE >= 026)	Display	
[XEQ] [ALPHA] RS [ALPHA]	*SEP T=?*	
100 [R/S]	*SEP P=?*	(absolute P)
125 [R/S]	*OIL G=?*	
30 [R/S]	*GAS G=?*	
.75 [R/S]	*GAS GS=0.7550*	
[R/S]*	*T=?*	
200 [R/S]	*RSI=?*	
350 [R/S]	*PBP=1954.3460*	PSI†
[R/S]*	*P=?*	
1200 [R/S]	*RSb=205.3054*	SCF/BBL†
[R/S]*	*P=?*	
2100 [R/S]	*RS=350.0000*	SCF/BBL†

```
GAS/OIL

SEP T=100.0000 F
SEP P=125.0000 PSI
OIL G=30.0000 API
GAS G=0.7500
GAS GS=0.7550
T=200.0000 F
RSI=350.0000 SCF/BBL
PBP=1954.3460 PSI
P=1200.0000 PSI

RSb=205.3054 SCF/BBL

P=2100.0000 PSI
RS=350.0000 SCF/BBL
```

*Press [R/S] if you are not using a printer.

†Press [ALPHA] to see the units if you are not using a printer.

For programmers who want to use Pac input and calculation subroutines in their own programs, the following examples illustrate the use of these routines. If you do not intend to use these subroutines in your programs, you should skip the next examples.

Example 2:

Assuming you have just run the last example, use the calculation subroutine to calculate the gas-oil ratio and its partial derivative at 2100 PSI if the molecular weight is 18.9. Use the gas gravity input routine (GASG) to input and store the new molecular weight.

Keystrokes	Display	
XEQ ALPHA GASG ALPHA	*GAS G=?*	
R/S	*MW=?*	
18.9 R/S	*0.6525*	GAS G
XEQ ALPHA CPBP ALPHA	*2219.6263*	PBP (PSI)

Since the pressure is less than the bubble point just calculated, use CRSb to calculate the gas-oil ratio. The routine requires the pressure in X.

2100 XEQ ALPHA CRSb ALPHA	*329.4223*	RSb (SCF/BBL)
R↓	*0.1716*	∂RSb/∂P
	MW=18.9000	

BUBBLE POINT PRESSURE

Given separator conditions, oil and gas gravity, temperature, and gas-oil ratio, the PBP program estimates the bubble point pressure (PBP).

Equations:

$$PBP = \left[\frac{RS}{A\ GAS\ GS \exp(C\ OIL\ G/T')} \right]^{1/B}$$

$$T' = T \text{ in } R$$

For OIL G $\leq$ 30 API:	For OIL G $>$ 30 API:
A = 0.0362	A = 0.0178
B = 1.0937	B = 1.1870
C = 25.7240	C = 23.9310

Calculation Subroutine:

The calculation subroutine for this program is called CPBP (*Calculate PBP*). The routine expects certain variables (SEP T, SEP P, OIL G, GAS G, T, RS) in the proper registers (see Appendix B). Upon return, PBP in PSI will be in X.

Range of Validity:

$$76 < SEP\ T < 150\ F$$
$$30 < SEP\ P < 535\ PSI$$

For 15.3 $<$ OIL G $\leq$ 30 API:	For 30.6 $<$ OIL G $<$ 59.5 API:
0.511 $<$ GAS G $<$ 1.351	0.530 $<$ GAS G $<$ 1.259

Reference:

Vasquez, M., and Beggs, H. D., "Correlations for Fluid Physical Property Predictions," *Journal of Petroleum Technology,* June, 1980, pp. 968-970.

User Instructions:

SIZE: 026

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1	Initialize program.		XEQ PBP	SEP T=?
2	Key in separator temperature.	SEP T	R/S	SEP P=?
3	Key in separator pressure.	SEP P	R/S	OIL G=?
4	Key in oil gravity.	OIL G	R/S	GAS G=?
5	Key in gas gravity and calculate GAS GS. If you do not know gas gravity, but do know molecular weight, press R/S to get the molecular weight prompt, and then key in molecular weight.	GAS G or MW	R/S R/S* R/S R/S R/S*	GAS GS= T=? MW=? GAS GS= T=?
6	Key in temperature.	T	R/S	RS=?
7	Key in gas-oil ratio and calculate PBP.	RS	R/S R/S*	PBP=† RS=?
8	For a new gas-oil ratio, go to step 7. For a new case, go to step 1. Key in only values which change. For values which do not change, ignore the prompts and press R/S.			

* Press R/S if you are not using a printer.

†Press ALPHA to see the units if you are not using a printer.

Example 1:

The gas-oil ratio of a 30 API oil is 350 SCF/BBL. The oil temperature is 200 F, and the gravity of the dissolved gas is 0.75. Separator conditions are 100 F and 125 PSI. What is the bubble point pressure of the oil? (These are the same values used in Example 1 of the *Gas-Oil Ratio* program. If you just ran that example, you can R/S past all the prompts.)

Keystrokes (SIZE >= 026)	Display	
XEQ ALPHA PBP ALPHA	*SEP T=?*	
100 R/S	*SEP P=?*	(absolute P)
125 R/S	*OIL G=?*	
30 R/S	*GAS G=?*	
.75 R/S	*GAS GS=0.7550*	

Keystrokes	Display	
R/S *	*T=?*	
200 R/S	*RS=?*	
350 R/S	*PBP=1954.3460*	PSI†

```
BUBBLE PT

SEP T=100.0000 F
SEP P=125.0000 PSI
OIL G=30.0000 API
GAS G=0.7500
GAS GS=0.7550
T=200.0000 F
RS=350.0000 SCF/BBL

PBP=1954.3460 PSI
```

For programmers who want to use Pac input and calculation subroutines in their own programs, the following example illustrates the use of these routines. If you do not intend to use these subroutines in your programs, you should skip the next example.

Example 2:

Assuming you have just run the last example, use the calculation subroutine to calculate the bubble point if the gas-oil ratio is 600 SCF/BBL. Use the gas-oil ratio input routine (IRS) to input and store the new gas-oil ratio. (Note that this input subroutine is called IRS to avoid conflict with the RS program.)

Keystrokes	Display	
XEQ ALPHA IRS ALPHA	*RS=?*	
600 R/S	*600.0000*	RS (SCF/BBL)
XEQ ALPHA CPBP ALPHA	*3199.1174*	PBP (PSI)

```
RS=600.0000 SCF/BBL
```

*Press R/S if you are not using a printer.

†Press ALPHA to see the units if you are not using a printer.

Notes

TWO-PHASE FORMATION VOLUME FACTOR

The BT program estimates the two-phase formation volume factor at the bubble point (BTBP), above the bubble point (BT), and below the bubble point (BTb). The two-phase formation volume factor is a combination of the individual gas and oil formation volume factors. For use in the Pac, BTBP, BT, and BTb are defined as the ratio of the volume at reservoir conditions to the volume at stock tank conditions.

Equations:

Above Bubble Point:

$$BT = BTBP \exp[CO\,(PBP - P)]$$

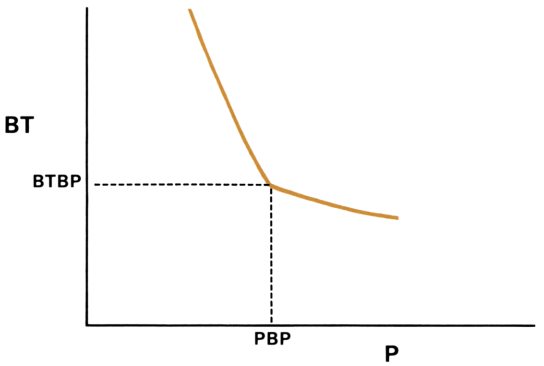
Below Bubble Point:

$$BTb = BOb + (RSI - RSb)\,BG'$$

At Bubble Point:

$$BTBP = BTb \text{ at } P = PBP \text{ (i.e., } RSb = RSI)$$

$$BG' = BG \text{ in BBL/SCF}$$



Calculation Subroutines:

There are two calculation subroutines for this program. The first subroutine is called CBTb (*Calculate BT Below Bubble Point*). The routine expects certain variables (Tc, Pc, STD T, STD P, SEP T, SEP P, OIL G, GAS G, T) in the proper registers (see Appendix B), and uses certain scratch registers (see Appendix A). In addition, a pressure must be in X. Upon return, BTb or BTBP will be in X, depending on whether P or PBP, respectively, was in X when the routine was called. RSb in SCF/BBL will be in Y.

The second subroutine is called CBT (*Calculate BT Above Bubble Point*). The routine expects certain variables (SEP T, SEP P, OIL G, GAS G, T, RSI, PBP, P) in the proper registers (see Appendix B). In addition, BTBP must be in X. Upon return, BT will be in X. Note that above the bubble point, BT = BO.

Range of Validity:

$$76 < \text{SEP T} < 150 \text{ F}$$
$$30 < \text{SEP P} < 535 \text{ PSI}$$
$$15.3 < \text{OIL G} < 59.5 \text{ API}$$

Above Bubble Point:

$$0.511 < \text{GAS G} < 1.351$$
$$111 < \text{P} < 9485 \text{ PSI}$$

At or Below Bubble Point:

For $15.3 < \text{OIL G} \leq 30$ API:	For $30.6 < \text{OIL G} < 59.5$ API:
$0.511 < \text{GAS G} < 1.351$	$0.530 < \text{GAS G} < 1.259$
$14.7 < \text{P} < 4542$ PSI	$14.7 < \text{P} < 6025$ PSI

At or below the bubble point, see also the "Range of Validity" section of the *Z Factor* program.

Remarks:

The correlations used for estimating RSb and BOb do not take into account the discontinuity at the bubble point. Consequently, the derivatives ∂RSb/∂P, ∂BOb/∂RSb, and ∂BOb/∂P do not accurately reflect the behavior of RSb and BOb approaching the bubble point. Because of this, BTb may be larger than expected at pressures slightly below the bubble point.

If you know that the pressures of interest are greater than the bubble point, you do not have to input values for Tc, Pc, STD T, or STD P. Just R/S past those prompts.

References:

Ramey, H. J., "Rapid Methods for Estimating Reservoir Compressibilities," *Journal of Petroleum Technology*, April, 1964, pp. 447-454.

Vasquez, M., and Beggs, H. D., "Correlations for Fluid Physical Property Predictions," *Journal of Petroleum Technology*, June, 1980, pp. 968-970.

User Instructions:

				SIZE: 026
STEP	**INSTRUCTIONS**	**INPUT**	**FUNCTION**	**DISPLAY**
1	Initialize program.		XEQ BT	Tc=?
2	Key in pseudocritical temperature.	Tc	R/S	Pc=?
3	Key in pseudocritical pressure.	Pc	R/S	STD T=?
4	Key in temperature at standard conditions. If the previously stored STD T is 0 F, it will be replaced by 60 F before this prompt appears.	STD T	R/S	STD P=?
5	Key in pressure at standard conditions. If the previously stored STD P is 0 PSI, it will be replaced by 14.65 PSI before this prompt appears.	STD P	R/S	SEP T=?
6	Key in separator temperature.	SEP T	R/S	SEP P=?
7	Key in separator pressure.	SEP P	R/S	OIL G=?
8	Key in oil gravity.	OIL G	R/S	GAS G=?
9	Key in gas gravity and calculate GAS GS. If you do not know gas gravity, but do know molecular weight, press R/S to get the molecular weight prompt, and then key in molecular weight.	GAS G or MW	R/S R/S* R/S R/S R/S*	GAS GS= T=? MW=? GAS GS= T=?
10	Key in temperature.	T	R/S	RSI=?
11	Key in initial gas-oil ratio and calculate PBP and BTBP.	RSI	R/S R/S* R/S*	PBP=† BTBP= P=?
12	Key in pressure. If the pressure is above the bubble point, BT will be calculated. If the pressure is below the bubble point, RSb for that pressure will be calculated, followed by BTb.	P P	R/S R/S* R/S R/S* R/S*	BT= P=? RSb=† BTb= P=?
13	For a new pressure, go to step 12. For a new case, go to step 1. Key in only values which change. For values which do not change, ignore the prompts and press R/S.			

* Press R/S if you are not using a printer.

† Press ALPHA to see the units if you are not using a printer.

Example 1:

A reservoir oil has a gravity of 39.8 API and an initial gas-oil ratio of 460 SCF/BBL at a temperature of 155 F. The dissolved gas has a gravity of 0.83, Tc of 429 R, and Pc of 664 PSI. (Tc and Pc were calculated using the *Pseudocritical Temperature and Pressure From Gas Gravity* program at a gas gravity of 0.83 and no sour gases.) Separator conditions are 100 F and 125 PSI, and the standard conditions for the gas are 60 F and 14.65 PSI. What is the two-phase formation volume factor at 1000 and 2000 PSI?

Keystrokes (SIZE >= 026)	**Display**	
[XEQ] [ALPHA] BT [ALPHA]	*Tc=?*	
429 [R/S]	*Pc=?*	
664 [R/S]	*STD T=?*	
60 [R/S]	*STD P=?*	
14.65 [R/S]	*SEP T=?*	
100 [R/S]	*SEP P=?*	(absolute P)
125 [R/S]	*OIL G=?*	
39.8 [R/S]	*GAS G=?*	
.83 [R/S]	*GAS GS=0.8373*	
[R/S] *	*T=?*	
155 [R/S]	*RSI=?*	
460 [R/S]	*PBP=1641.5439*	PSI†
[R/S] *	*BTBP=1.2673*	
[R/S] *	*P=?*	
1000 [R/S]	*RSb=255.4189*	SCF/BBL†
[R/S] *	*BTb=1.6950*	
[R/S] *	*P=?*	
2000 [R/S]	*BT=1.2604*	

*Press [R/S] if you are not using a printer.

†Press [ALPHA] to see the units if you are not using a printer.

```
2PH VOL FACT

Tc=429.0000 R
Pc=664.0000 PSI
STD T=60.0000 F
STD P=14.6500 PSI
SEP T=100.0000 F
SEP P=125.0000 PSI
OIL G=39.8000 API
GAS G=0.8300
GAS GS=0.8373
T=155.0000 F
RSI=460.0000 SCF/BBL
PBP=1641.5439 PSI

BTBP=1.2673

P=1000.0000 PSI
RSb=255.4189 SCF/BBL
BTb=1.6950

P=2000.0000 PSI
BT=1.2604
```

For programmers who want to use Pac input and calculation subroutines in their own programs, the following example illustrates the use of these routines. If you do not intend to use these subroutines in your programs, you should skip the next example.

Example 2:

Assuming you have just run the last example, use the calculation subroutines to compute the formation volume factor if the separator was at 280 F. Use the separator pressure and temperature input routine (SEPTP) to input and store the new separator conditions.

Keystrokes	Display	
XEQ ALPHA SEPTP ALPHA	*SEP T=?*	
280 R/S	*SEP P=?*	
XEQ ALPHA CPBP ALPHA	*1620.1684*	PBP (PSI)

Since the last pressure used in Example 1 is greater than the bubble point just calculated, use CBT to calculate the formation volume factor. This routine requires BTBP in X, which is calculated using CBTb with PBP in X. (BTBP can also be calculated using CBOb if Tc, Pc, STD T, and STD P are not known.)

Keystrokes	Display	
XEQ ALPHA CBTb ALPHA	*1.2665*	BTBP
XEQ ALPHA CBT ALPHA	*1.2596*	BT

Now use the P input routine to input and store the 1000 PSI pressure from Example 1. Since this pressure is less than the bubble point just calculated, use CBTb to calculate the formation volume factor.

XEQ ALPHA P ALPHA	*P=?*	
1000 R/S	*1000.0000*	P (PSI)
XEQ ALPHA CBTb ALPHA	*1.6858*	BTb
	SEP T=280.0000 F P=1000.0000 PSI	

WATER ISOTHERMAL COMPRESSIBILITY

Given salinity, temperature, and pressure, the CW program estimates the isothermal coefficient of compressibility for reservoir brines (CW). The program includes an option to correct the compressibility for gas saturation.

Equations:

$$CW\,(H_2O) = (A + BT + CT^2)/10^6$$

$$A = 3.8546 - 0.000134\,P$$

$$B = -0.01052 + 4.77\,(10^{-7})\,P$$

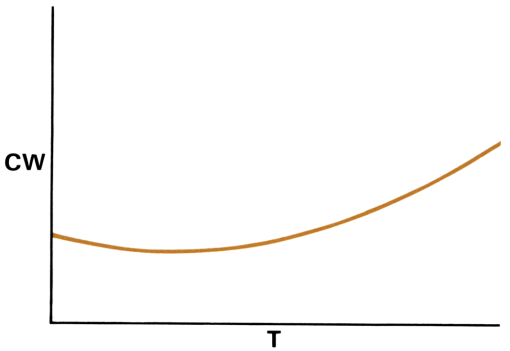
$$C = 3.9267\,(10^{-5}) - 8.8\,(10^{-10})\,P$$

Gas Saturation Correction:

$$CW\,(gas) = CW\,(H_2O)\,[1 + 8.9\,(10^{-3})\,RSW]$$

Salinity Correction:

$$CW\,(brine) = CW\left(\begin{matrix}H_2O\\ or\\ gas\end{matrix}\right)\{[-0.052 + 2.7\,(10^{-4})\,T - 1.14\,(10^{-6})\,T^2 + 1.121\,(10^{-9})\,T^3]\,\%NACL^{0.7} + 1\}$$



Calculation Subroutine:

The calculation subroutine for this program is called CCW (*Calculate CW*). The routine expects certain variables (%NACL, T, P) in the proper registers (see Appendix B), and uses certain scratch registers and flags (see Appendix A). Upon return, CW in 1/PSI will be in X.

Range of Validity:

$$80 < T < 250 \text{ F}$$
$$1000 < P < 6000 \text{ PSI}$$
$$0 \leq \%\text{NACL} < 25$$

References:

Craft, B. C. and Hawkins, M. F., *Applied Petroleum Reservoir Engineering,* Prentice-Hall, 1959, p. 130.

Meehan, D. N., "A Correlation for Water Compressibility," *Petroleum Engineer,* November, 1980, pp. 125-126.

Numbere, D., Brigham, W. E., and Standing, M. B., *Correlations for Physical Properties of Petroleum Reservoir Brines,* Petroleum Research Institute, Stanford University, November, 1977, p. 17.

User Instructions:

SIZE: 020

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1	Initialize program.		XEQ CW	RSW>0? Y/N:‡
2	If the water is gas-saturated, answer "Y". If not, answer "N".	Y or N	R/S	%NACL=?
3	Key in weight percent sodium chloride. If you do not know weight percent, but do know parts per million, press R/S to get the parts per million prompt, and then key in parts per million.	%NACL	R/S	T=?
		or		
			R/S	PPM=?
		PPM	R/S	T=?
4	Key in temperature.	T	R/S	P=?
5	Key in pressure and calculate CW.	P	R/S	CW=†
			R/S*	P=?
6	For a new pressure, go to step 5. For a new case, go to step 1. Key in only values which change. For values which do not change, ignore the prompts and press R/S.			
	*Press R/S if you are not using a printer.			
	†Press ALPHA to see the units if you are not using a printer.			
	‡The last character of this prompt will be Y if the prompt is currently true, or N if the prompt is currently false.			

Example 1:

A reservoir brine has 30,000 PPM sodium chloride. The brine is at a temperature of 200 F and a pressure of 3000 PSI. What is the isothermal compressibility of the brine? Determine the compressibility for both a gas-saturated and a gas-free brine.

Keystrokes (SIZE >= 020)	Display	
[XEQ] [ALPHA] CW [ALPHA]	*RSW>0? Y/N:‡*	
Y [R/S]	*%NACL=?*	
[R/S]	*PPM=?*	
30000 [R/S]	*T=?*	
200 [R/S]	*P=?*	
3000 [R/S]	*CW=3.2549E-6*	1/PSI†
[XEQ] [ALPHA] CW [ALPHA]	*RSW>0? Y/N:Y*	
N [R/S]	*%NACL=?*	
[R/S]	*PPM=?*	
[R/S]	*T=?*	
[R/S]	*P=?*	
[R/S]	*CW=2.8682E-6*	1/PSI†

```
H2O ISO CMP

RSW>0: YES
PPM=30000.0000
T=200.0000 F
P=3000.0000 PSI

CW=3.2549E-6 1/PSI


H2O ISO CMP

RSW>0: NO

CW=2.8682E-6 1/PSI
```

For programmers who want to use Pac input and calculation subroutines in their own programs, the following example illustrates the use of these routines. If you do not intend to use these subroutines in your programs, you should skip the next example.

†Press [ALPHA] to see the units if you are not using a printer.

‡The last character of this prompt will be Y if the prompt is currently true, or N if the prompt is currently false.

Example 2:

Assuming you have just run the last example, use the calculation subroutine to compute the isothermal compressibility in 1/PSI for the same brine with 200,000 PPM sodium chloride, assuming the brine is gas-saturated. Use the sodium chloride input routine (%NACL) to input and store the new PPM.

Since this example is for a gas-saturated brine, the status set by answering N to the ***RSW>0? Y/N*** question is no longer valid. The appendices describe how to set the proper status. From Appendix A, flag 06 is needed for this calculation subroutine. From Appendix B, the flag is used to distinguish between gas-free and gas-saturated water, and must be set if the water is gas-saturated.

Keystrokes	Display	
[SF] 06		(gas-saturated)
[XEQ] [ALPHA] %NACL [ALPHA]	***%NACL=?***	
[R/S]	***PPM=?***	
2 [EEX] 5 [R/S]	***20.0000***	%NACL
[XEQ] [ALPHA] CCW [ALPHA]	***2.5259 -06***	CW (1/PSI)

(Remember that all input and calculation subroutines return values in Pac English default units.)

```
PPM=200000.0000
```

WATER FORMATION VOLUME FACTOR

The BW program estimates the formation volume factor for reservoir brines (BW). The program includes an option to correct the formation volume factor for gas saturation. For use in the Pac, BW is defined as the ratio of the volume at reservoir conditions to the volume at stock tank conditions.

Equations:

$$BW\,(H_2O) = A + BP + CP^2$$

Gas-Free Water:

$$A = 0.9947 + 5.8\,(10^{-6})\,T + 1.02\,(10^{-6})\,T^2$$

$$B = -4.228\,(10^{-6}) + 1.8376\,(10^{-8})\,T - 6.77\,(10^{-11})\,T^2$$

$$C = 1.3\,(10^{-10}) - 1.3855\,(10^{-12})\,T + 4.285\,(10^{-15})\,T^2$$

Gas-Saturated Water:

$$A = 0.9911 + 6.35\,(10^{-5})\,T + 8.5\,(10^{-7})\,T^2$$

$$B = -1.093\,(10^{-6}) - 3.497\,(10^{-9})\,T + 4.57\,(10^{-12})\,T^2$$

$$C = -5\,(10^{-11}) + 6.429\,(10^{-13})\,T - 1.43\,(10^{-15})\,T^2$$

Salinity Correction:

$$\begin{aligned} BW\,(\text{brine}) = BW\,(H_2O)\,[\{&5.1\,(10^{-8})\,P \\ &+ [5.47\,(10^{-6}) - 1.95\,(10^{-10})\,P]\,(T - 60) \\ &+ [-3.23\,(10^{-8}) + 8.5\,(10^{-13})\,P]\,(T - 60)^2\}\,\%NACL + 1] \end{aligned}$$

Calculation Subroutine:

The calculation subroutine for this program is called CBW (*Calculate BW*). The routine expects certain variables (%NACL, T, P) in the proper registers (see Appendix B), and uses certain flags (see Appendix A). Upon return, BW will be in X.

Range of Validity:

$$100 < T < 250\ F$$
$$1000 < P < 5000\ PSI$$
$$0 \leqslant \%NACL < 25$$

References:

Craft, B. C. and Hawkins, M. F., *Applied Petroleum Reservoir Engineering,* Prentice-Hall, 1959, p. 131.

Numbere, D., Brigham, W. E., and Standing, M. B., *Correlations for Physical Properties of Petroleum Reservoir Brines,* Petroleum Research Institute, Stanford University, November, 1977, p. 16.

Ramey, H. J., Stanford University, unpublished correspondence.

User Instructions:

SIZE: 020

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1	Initialize program.		XEQ BW	RSW>0? Y/N:‡
2	If the water is gas-saturated, answer "Y". If not, answer "N".	Y or N	R/S	%NACL=?
3	Key in weight percent sodium chloride. If you do not know weight percent, but do know parts per million, press R/S to get the parts per million prompt, and then key in parts per million.	%NACL or PPM	R/S R/S R/S	T=? PPM=? T=?
4	Key in temperature.	T	R/S	P=?
5	Key in pressure and calculate BW.	P	R/S R/S*	BW= P=?
6	For a new pressure, go to step 5. For a new case, go to step 1. Key in only values which change. For values which do not change, ignore the prompts and press R/S.			

* Press R/S if you are not using a printer.

‡The last character of this prompt will be Y if the prompt is currently true, or N if the prompt is currently false.

Example 1:

A reservoir water is at 100 F and 5000 PSI. If the water is gas-saturated, what is the formation volume factor? If the water is gas-free?

Keystrokes (SIZE >= 020)	Display
XEQ ALPHA BW ALPHA	*RSW>0? Y/N:‡*
Y R/S	*%NACL=?*
0 R/S	*T=?*
100 R/S	*P=?*
5000 R/S	*BW=0.9990*
XEQ ALPHA BW ALPHA	*RSW>0? Y/N:Y*
N R/S	*%NACL=?*
R/S	*PPM=?*
R/S	*T=?*
R/S	*P=?*
R/S	*BW=0.9910*

```
H2O VOL FACT

RSW>0: YES
%NACL=0.0000
T=100.0000 F
P=5000.0000 PSI

BW=0.9990


H2O VOL FACT

RSW>0: NO

BW=0.9910
```

For programmers who want to use Pac input and calculation subroutines in their own programs, the following example illustrates the use of these routines. If you do not intend to use these subroutines in your programs, you should skip the next example.

‡The last character of this prompt will be Y if the prompt is currently true, or N if the prompt is currently false.

Example 2:

Assuming you have just run the last example, use the calculation subroutine to compute the formation volume factor for the same water at a temperature of 200 F if the water is gas-free. Use the temperature input routine (T) to input and store the new T.

Since the calculation in the last example was for gas-free water, the status set by answering N to the ***RSW>0? Y/N*** question is still valid for use by the calculation subroutine.

Keystrokes	Display	
XEQ ALPHA T ALPHA	*T=?*	
200 R/S	*200.0000*	T (F)
XEQ ALPHA CBW ALPHA	*1.0210*	BW
	T=200.0000 F	

WATER VISCOSITY

The UW program estimates the viscosity of reservoir brines (UW) as a function of salinity, temperature, and pressure. The program also estimates the saturation pressure of water (PSAT) as a function of temperature.

Equations:

$$\begin{aligned} UW\,(H_2O) = {} & 241.4\,(10^{-4})\,10^{[247.8\,/\,(T'\,-\,140)]} \\ & \cdot [1 + (P' - PSAT')\,1.0467\,(10^{-6})\,(T' - 305)] \end{aligned}$$

$$T' = T \text{ in } K$$

$$P' = P \text{ in BAR}$$

$$PSAT' = PSAT \text{ in BAR}$$

Salinity Correction:

$$\begin{aligned} UW\,(\text{brine}) = {} & UW\,(H_2O)\,\{1 - 1.87\,(10^{-3})\,\%NACL^{0.5} \\ & + 2.18\,(10^{-4})\,\%NACL^{2.5} + (T^{0.5} - 0.0135\,T) \\ & \cdot [2.76\,(10^{-3})\,\%NACL - 3.44\,(10^{-4})\,\%NACL^{1.5}]\} \end{aligned}$$

Saturation Pressure:

$$PSAT' = 22088 \exp\left[\frac{374.136 - T''}{T'} \sum_{i=1}^{8} A_i (0.65 - 0.01\,T'')^{i-1}\right]$$

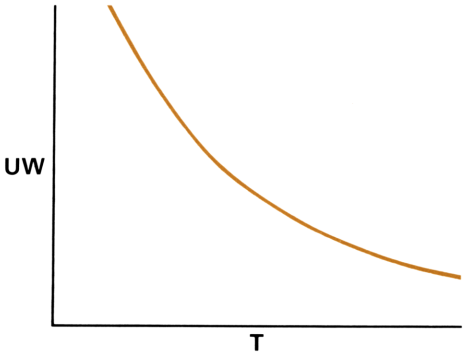
$$PSAT' = PSAT \text{ in KPA}$$

$$T' = T \text{ in K}$$

$$T'' = T \text{ in C}$$

PSAT Coefficients

i	A_i
1	−7.419242
2	−0.29721
3	−0.1155286
4	−0.008685635
5	0.001094098
6	0.00439993
7	0.002520658
8	0.0005218684



Calculation Subroutines:

There are two calculation subroutines for this program. The first subroutine is called CUW (*Calculate UW*). The routine expects certain variables (%NACL, T, P) in the proper registers (see Appendix B). Upon return, UW in CP will be in X.

The second subroutine is called CPSAT (*Calculate PSAT*). The routine expects T in X. Upon return, PSAT in PSI will be in X.

Range of Validity:

For UW:

$$32 < T < 572 \text{ F}$$
$$PSAT < P < 11600 \text{ PSI}$$
$$0 \leqslant \%NACL < 25$$

For PSAT:

$$32 < T < 705 \text{ F}$$
$$0.0887 < P < 3203.6 \text{ PSI}$$

The program will halt and display ***P < PSAT*** for pressures less than the saturation pressure of water for the temperature being used. If this occurs, press [←] to see the saturation pressure. Then restart the program using either a pressure higher than PSAT or a lower temperature.

Remarks:

The saturation pressure of water is the pressure at which water will vaporize for a given temperature, and this temperature is called the saturation temperature. If the pressure is less than PSAT, a vapor phase is present, and the liquid viscosity cannot be calculated.

References:

Haywood, R. W., "Sixth International Conference on the Properties of Steam—Supplement on Transport Properties," *J. Eng. Power, Trans. ASME,* January, 1966, pp. 63-66.

Numbere, D., Brigham, W. E., and Standing, M. B., *Correlations for Physical Properties of Petroleum Reservoir Brines,* Petroleum Research Institute, Stanford University, November, 1977, p. 8.

Keenan, J. H., *et. al, Steam Tables,* John Wiley and Sons, 1969, p. 141.

Van Wylen, G. J., and Sonntag, R. E., *Fundamentals of Classical Thermodynamics,* John Wiley and Sons, 1973, pp. 40-41.

User Instructions:

SIZE: 020

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1	Initialize program.		[XEQ] UW	%NACL=?
2	Key in weight percent sodium chloride. If you do not know weight percent, but do know parts per million, press [R/S] to get the parts per million prompt, and then key in parts per million.	%NACL or PPM	[R/S] [R/S] [R/S]	T=? PPM=? T=?
3	Key in temperature.	T	[R/S]	P=?
4	Key in pressure and calculate UW.	P	[R/S] [R/S]*	UW=† P=?
5	For a new pressure, go to step 4. For a new case, go to step 1. Key in only values which change. For values which do not change, ignore the prompts and press [R/S].			

* Press [R/S] if you are not using a printer.

†Press [ALPHA] to see the units if you are not using a printer.

Example 1:

A reservoir brine has 18 percent sodium chloride. The brine is at a temperature of 50 C and a pressure of 180 ATM. What is the viscosity of the brine in PA*S? If the pressure doubles, what is the viscosity? Turn on the output portion of the Unit Management System (set flag 10) to allow you to change the output units.

Keystrokes (SIZE >= 020)	Display	
[f] [SF] 10 [f] [ENG] 4		
[XEQ] [ALPHA] UW [ALPHA]	*%NACL=?*	
18 [R/S]	*T=?*	
50 [ALPHA] C [R/S]	*P=?*	
180 [ALPHA] ATM [R/S]	*UW, CP?*	
PA*S [R/S]	*UW=825.48E-6*	PA*S†
[R/S] *	*P=?*	
[←]	*2.6453 03*	

Pressing [←] shows you the previous value for P. Whenever there is an input or output prompt, the number in X always corresponds to the units shown in the ALPHA register. Check the ALPHA register to see what the units are.

[ALPHA]	*PSI*	

The number in X is 180 ATM converted to PSI (English default units because flag 09 is clear).

[ALPHA] 2 [×] [R/S]	*UW, PA*S?*	
[R/S]	*UW=828.33E-6*	PA*S†

Notice that the units you selected for UW, the primary output variable, were saved for you.

*Press [R/S] if you are not using a printer.

†Press [ALPHA] to see the units if you are not using a printer.

```
H2O VIS

%NACL=18.000E0
T=50.000E0 C
P=180.00E0 ATM

UW=825.48E-6 PA*S

P=5.2905E3 PSI
UW=828.33E-6 PA*S
```

For programmers who want to use Pac input and calculation subroutines in their own programs, the following examples illustrate the use of these routines. If you do not intend to use these subroutines in your programs, you should skip the next examples.

Example 2:

Assuming you have just run the last example, use the calculation subroutine to compute the viscosity in PA*S for salt-free water at the same temperature and pressure. Use the sodium chloride input routine (%NACL) to input and store the new %NACL.

Keystrokes	Display	
XEQ ALPHA %NACL ALPHA	*%NACL=?*	
0 R/S	*0.0000 00*	%NACL
XEQ ALPHA CUW ALPHA	*547.93 -03*	UW (CP)
ALPHA	*F*	

All calculation subroutines return values in Pac English default units. Because of this, the contents of the ALPHA register after executing a calculation subroutine are not related to the units of the variables that were calculated.

PA*S-CP ALPHA	*547.93 -03*	
XEQ ALPHA INCON ALPHA	*547.93 -06*	UW (PA*S)
	%NACL=0.0000E0	

Example 3:

For a temperature of 120 C, what is the saturation pressure of water in PSI? Remember that all inputs to a calculation subroutine must be in Pac English default units.

Keystrokes	Display	
120 [ALPHA] C-F [ALPHA]	*120.00 00*	T (C)
[XEQ] [ALPHA] CON [ALPHA]	*248.00 00*	T (F)
[XEQ] [ALPHA] CPSAT [ALPHA]	*28.793 00*	PSAT (PSI)
■ [CF] 10 ■ [FIX] 4	*28.7929*	

GAS-WATER RATIO

Given salinity, temperature, and pressure, the RSW program estimates the dissolved gas-water ratio (RSW) for reservoir brines.

Equations:

$$\text{RSW}\,(H_2O) = A + BP + CP^2$$

$$\partial\text{RSW}/\partial\text{P}\,(H_2O) = B + 2CP$$

$$A = 2.12 + 3.45\,(10^{-3})\,T - 3.59\,(10^{-5})\,T^2$$

$$B = 0.0107 - 5.26\,(10^{-5})\,T + 1.48\,(10^{-7})\,T^2$$

$$C = -8.75\,(10^{-7}) + 3.9\,(10^{-9})\,T - 1.02\,(10^{-11})\,T^2$$

Salinity Correction:

$$\text{S.C.} = 1 - [7.53\,(10^{-2}) - 1.73\,(10^{-4})\,T]\,\%\text{NACL}$$

$$\text{RSW (brine)} = \text{RSW}\,(H_2O) \cdot \text{S.C.}$$

$$\partial\text{RSW}/\partial\text{P (brine)} = \partial\text{RSW}/\partial\text{P}\,(H_2O) \cdot \text{S.C.}$$

Calculation Subroutine:

The calculation subroutine for the gas-water ratio program is called CRSW (*Calculate RSW*). The routine expects certain variables (%NACL, T, P) in the proper registers (see Appendix B), and uses certain scratch registers (see Appendix A). Upon return, RSW in SCF/BBL will be in X, the partial derivative of RSW with respect to P at constant T (∂RSW/∂P) will be in Y, and the salinity correction (S.C.) will be in Z.

Range of Validity:

$$90 < T < 250\ \text{F}$$
$$500 < P < 5000\ \text{PSI}$$
$$0 \leqslant \%\text{NACL} < 3$$

References:

Craft, B. C., and Hawkins, M. F., *Applied Petroleum Reservoir Engineering,* Prentice-Hall, 1959, p. 130.

Ramey, H. J., Stanford University, unpublished correspondence.

User Instructions:

				SIZE: 020
STEP	**INSTRUCTIONS**	**INPUT**	**FUNCTION**	**DISPLAY**
1	Initialize program.		XEQ RSW	%NACL=?
2	Key in weight percent sodium chloride. If you do not know weight percent, but do know parts per million, press R/S to get the parts per million prompt, and then key in parts per million.	%NACL or PPM	R/S R/S R/S	T=? PPM=? T=?
3	Key in temperature.	T	R/S	P=?
4	Key in pressure and calculate RSW.	P	R/S R/S*	RSW=† P=?
5	For a new pressure, go to step 4. For a new case, go to step 1. Key in only values which change. For values which do not change, ignore the prompts and press R/S. *Press R/S if you are not using a printer. †Press ALPHA to see the units if you are not using a printer.			

Example 1:

Determine the gas-water ratio of a reservoir brine containing 25,000 PPM sodium chloride. The brine is at a temperature of 150 F and a pressure of 3100 PSI.

Keystrokes (SIZE >= 020)

XEQ ALPHA RSW ALPHA
R/S
25000 R/S
150 R/S
3100 R/S

Display

%NACL=?
PPM=?
T=?
P=?
RSW=13.9132 SCF/BBL†

```
GAS/WATER

PPM=25000.0000
T=150.0000 F
P=3100.0000 PSI

RSW=13.9132 SCF/BBL
```

†Press ALPHA to see the units if you are not using a printer.

For programmers who want to use Pac calculation subroutines in their own programs, the following example illustrates the use of these routines. If you do not intend to use these subroutines in your programs, you should skip the next example.

Example 2:

Assuming you have just run the last example, use the calculation subroutine to recompute the gas-water ratio, its partial derivative, and the salinity correction.

Keystrokes	Display	
XEQ ALPHA CRSW ALPHA	*13.9132*	RSW (SCF/BBL)
R↓	*0.0026*	∂RSW/∂P
R↓	*0.8766*	S.C.

Notes

ROCK COMPRESSIBILITY

The CFR program estimates the formation or rock compressibility (CFR) given the percent porosity. The technique used is based on Hall's correlation for rock compressibility.

Equations:

$$CFR = 1.87\,(10^{-6})\,(\%POR/100)^{-0.415}$$

Calculation Subroutine:

The calculation subroutine for this program is called CCFR (*Calculate CFR*). The routine expects certain variables (%POR) in the proper registers (see Appendix B). Upon return, CFR in 1/PSI will be in X.

Range of Validity:

$$2 < \%POR < 26$$

Remarks:

This technique does not provide valid approximations to compressibility for unconsolidated or friable sandstones.

References:

Hall, H. N., "Compressibility of Reservoir Rocks," *Trans. AIME,* 1953, 198, pp. 309-311.

Craft, B. C., and Hawkins, M. F., *Applied Petroleum Reservoir Engineering,* Prentice-Hall, 1959, p. 132.

Earlougher, R. C., *Advances in Well Test Analysis,* Society of Petroleum Engineers of AIME, 1977, pp. 229-230.

Meehan, D. N., Champlin Petroleum Company, unpublished correspondence.

User Instructions:

				SIZE: 019
STEP	**INSTRUCTIONS**	**INPUT**	**FUNCTION**	**DISPLAY**
1	Initialize program.		XEQ CFR	%POR=?
2	Key in percent rock porosity and calculate CFR.	%POR	R/S R/S*	CFR=† %POR=?
3	For a new percent porosity, go to step 2. For a new case, go to step 1. Key in only values which change. For values which do not change, ignore the prompts and press R/S. *Press R/S if you are not using a printer. †Press ALPHA to see the units if you are not using a printer.			

Example 1:

The porosity of a certain reservoir is 20 percent. What is its approximate compressibility?

Keystrokes (SIZE >= 019)	Display	
XEQ ALPHA CFR ALPHA	*%POR=?*	
20 R/S	*CFR=3.6468E-6*	1/PSI†

```
ROCK CMP

%POR=20.0000

CFR=3.6468E-6 1/PSI
```

For programmers who want to use Pac input and calculation subroutines in their own programs, the following example illustrates the use of these routines. If you do not intend to use these subroutines in your programs, you should skip the next example.

†Press ALPHA to see the units if you are not using a printer.

Example 2:

With the calculation subroutine, estimate the rock compressibility in 1/PSI for a reservoir rock with 14 percent porosity. Use the porosity input routine (%POR) to input and store the new %POR.

Keystrokes	Display	
XEQ ALPHA %POR ALPHA	*%POR=?*	
14 R/S	*14.0000*	%POR
XEQ ALPHA CCFR ALPHA	*4.2286 -06*	CFR (1/PSI)
	%POR=14.0000	

Notes

TOTAL ISOTHERMAL COMPRESSIBILITY

The CT program estimates the total isothermal coefficient of compressibility of a formation above and below the bubble point (CT and CTb). The effects of all three reservoir fluids (gas, oil, and brine) and of the formation are included. The effect of dissolved gas on the oil and brine is taken into account. CT and CTb are calculated by a weighted sum of the compressibilities of the separate phases. The volumetric phase saturations are the weighting factors in the sum.

Equations:

Oil Present:

Above Bubble Point:

$$CT = SO\ CO + SW\ CW + CFR$$

Below Bubble Point:

$$CTb = SO\ COb + SG\ CG + SW\ CW + CFR$$

No Oil Present:

$$CT = SG\ CG + SW\ CW + CFR$$

$$SO = \frac{\%SO}{100},\ SW = \frac{\%SW}{100},\ SG = \frac{\%SG}{100}$$

Calculation Subroutines:

There are two calculation subroutines for this program. Both routines expect certain input variables (Pc, SEP T, SEP P, OIL G, GAS G, %NACL, %POR, %SO, %SW, T, P) in the proper registers (see Appendix B), and use certain scratch registers and flags (see Appendix A). In addition, TR must be in Z, PR must be in Y, and P must be in X.

The first subroutine, CCTb (*Calculate CT Below Bubble Point*), expects certain additional variables (STD T, STD P) in the proper registers. Upon return, CTb in 1/PSI will be in X, and RSb in SCF/BBL will be in Y. The second subroutine, CCT (*Calculate CT Above Bubble Point*), expects an additional variable (RSI) in the proper register. Upon return, CT in 1/PSI will be in X.

Range of Validity:

$$0 \leqslant \%SO \leqslant 100$$
$$0 \leqslant \%SW \leqslant 100$$
$$0 \leqslant \%SO + \%SW \leqslant 100$$

See also the "Range of Validity" section of the *Gas Isothermal Compressibility, Oil Isothermal Compressibility, Water Isothermal Compressibility,* and *Rock Compressibility* programs.

Remarks:

The correlations used for estimating COb do not take into account the discontinuity at the bubble point. Because of this, CTb may be larger than expected at pressures slightly below the bubble point.

If you know that the pressures of interest are greater than the bubble point, and there is oil present (%SO $>$ 0), you do not have to input values for Tc, Pc, STD T, or STD P. Just [R/S] past those prompts.

References:

See the "References" section of the *Gas Isothermal Compressibility, Oil Isothermal Compressibility, Water Isothermal Compressibility,* and *Rock Compressibility* programs.

User Instructions:

				SIZE: 026
STEP	**INSTRUCTIONS**	**INPUT**	**FUNCTION**	**DISPLAY**
1	Initialize program.		[XEQ] CT	RSW>0? Y/N:‡
2	If the water is gas-saturated, answer "Y". If not, answer "N".	Y or N	[R/S]	Tc=?
3	Key in pseudocritical temperature.	Tc	[R/S]	Pc=?
4	Key in pseudocritical pressure.	Pc	[R/S]	STD T=?
5	Key in temperature at standard conditions. If the previously stored STD T is 0 F, it will be replaced by 60 F before this prompt appears.	STD T	[R/S]	STD P=?
6	Key in pressure at standard conditions. If the previously stored STD P is 0 PSI, it will be replaced by 14.65 PSI before this prompt appears.	STD P	[R/S]	SEP T=?
7	Key in separator temperature.	SEP T	[R/S]	SEP P=?
8	Key in separator pressure.	SEP P	[R/S]	OIL G=?
9	Key in oil gravity.	OIL G	[R/S]	GAS G=?
	‡The last character of this prompt will be Y if the prompt is currently true, or N if the prompt is currently false.			

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
10	Key in gas gravity. If you do not know gas gravity, but do know molecular weight, press [R/S] to get the molecular weight prompt, and then key in molecular weight.	GAS G or MW	[R/S] [R/S] [R/S]	%NACL=? MW=? %NACL=?
11	Key in weight percent sodium chloride. If you do not know weight percent, but do know parts per million, press [R/S] to get the parts per million prompt, and then key in parts per million.	%NACL or PPM	[R/S] [R/S] [R/S]	%POR=? PPM=? %POR=?
12	Key in percent porosity.	%POR	[R/S]	T=?
13	Key in temperature.	T	[R/S]	RSI=?
14	Key in initial gas-oil ratio.	RSI	[R/S]	%SO=?
15	Key in volume percent oil.	%SO	[R/S]	%SW=?
16	Key in volume percent water and calculate %SG. If there is oil present, calculate GAS GS and PBP.**	%SW	[R/S] [R/S]* [R/S]* [R/S]*	%SG= GAS GS= PBP=† P=?
17	Key in pressure. If there is oil present and the pressure is above the bubble point, or if there is no oil present, CT will be calculated. If there is oil present and the pressure is below the bubble point, RSb for that pressure will be calculated, followed by CTb.	P P	[R/S] [R/S]* [R/S] [R/S]* [R/S]*	CT=† %SO=? RSb=† CTb=† %SO=?
18	For a new volume percent oil and water, and a new pressure, go to step 15. For a new case, go to step 1. Key in only values which change. For values which do not change, ignore the prompts and press [R/S].			
	* Press [R/S] if you are not using a printer.			
	†Press [ALPHA] to see the units if you are not using a printer.			
	** When CT is run, GAS GS and PBP will only be output the first time step 16 is reached.			

Example 1:

It is desired to estimate the total isothermal coefficient of compressibility of a particular oil zone. The formation has a porosity of 22 percent, and is at a temperature of 155 F. The fluid in the reservoir is 53 percent oil with 39.8 API gravity, 35 percent water with 10,000 PPM sodium chloride, and 12 percent gas with 0.83 gravity. Since the water is presumably in equilibrium with the hydrocarbons, it can be assumed to be gas-saturated.

At pressures above the bubble point, no free gas is present in the reservoir, leaving 65 percent oil and 35 percent water.

Using the *Pseudocritical Temperature and Pressure From Gas Gravity* program for a reservoir gas with 0.83 gravity and no sour gases, Tc of 429 R and Pc of 664 PSI are calculated. Standard conditions for the gas are 60 F and 14.65 PSI, and separator conditions are 100 F and 125 PSI. If the initial gas-oil ratio is 460 SCF/BBL, what is the total isothermal compressibility for the system in 1/PSI at 1000 and 2000 PSI?

Keystrokes (SIZE >= 026)	**Display**	
[ENG] 4		
[XEQ] [ALPHA] CT [ALPHA]	*RSW>0? Y/N:*‡	
Y [R/S]	*Tc=?*	
429 [R/S]	*Pc=?*	
664 [R/S]	*STD T=?*	
60 [R/S]	*STD P=?*	
14.65 [R/S]	*SEP T=?*	
100 [R/S]	*SEP P=?*	(absolute P)
125 [R/S]	*OIL G=?*	
39.8 [R/S]	*GAS G=?*	
.83 [R/S]	*%NACL=?*	
[R/S]	*PPM=?*	
10000 [R/S]	*%POR=?*	
22 [R/S]	*T=?*	
155 [R/S]	*RSI=?*	
460 [R/S]	*%SO=?*	
53 [R/S]	*%SW=?*	
35 [R/S]	*%SG=12.0000E0*	
[R/S] *	*GAS GS=837.29E-3*	
[R/S] *	*PBP=1.6415E3*	PSI†

*Press [R/S] if you are not using a printer.

†Press [ALPHA] to see the units if you are not using a printer.

‡The last character of this prompt will be Y if the prompt is currently true, or N if the prompt is currently false.

Keystrokes	Display	
R/S *	*P=?*	
1000 R/S	*RSb=255.42E0*	SCF/BBL†
R/S *	*CTb=433.73E-6*	1/PSI†
R/S *	*%SO=?*	
65 R/S	*%SW=?*	
R/S	*%SG=0.0000E0*	
R/S *	*P=?*	
2000 R/S	*CT=14.532E-6*	1/PSI†

```
TOT ISO CMP

RSW>0: YES
Tc=429.00E0 R
Pc=664.00E0 PSI
STD T=60.000E0 F
STD P=14.650E0 PSI
SEP T=100.00E0 F
SEP P=125.00E0 PSI
OIL G=39.800E0 API
GAS G=830.00E-3
PPM=10.000E3
%POR=22.000E0
T=155.00E0 F
RSI=460.00E0 SCF/BBL
%SO=53.000E0
%SW=35.000E0
%SG=12.000E0
GAS GS=837.29E-3
PBP=1.6415E3 PSI
P=1.0000E3 PSI
RSb=255.42E0 SCF/BBL

CTb=433.73E-6 1/PSI

%SO=65.000E0
%SG=0.0000E0
P=2.0000E3 PSI
CT=14.532E-6 1/PSI
```

*Press R/S if you are not using a printer.

†Press ALPHA to see the units if you are not using a printer.

For programmers who want to use Pac input and calculation subroutines in their own programs, the following example illustrates the use of these routines. If you do not intend to use these subroutines in your programs, you should skip the next example.

Example 2:

Assuming you have just run the last example, use the calculation subroutines to compute the isothermal compressibility in 1/PSI at the pressures in the example if the standard pressure used is 15.025 PSI. Use the standard temperature and pressure input routine (STDTP) to input and store the new STD P.

Keystrokes	Display	
XEQ ALPHA STDTP ALPHA	*STD T=?*	
R/S	*STD P=?*	
15.025 R/S	*15.025 00*	STD P (PSI)

Since there is oil present, and the last pressure used in Example 1 is greater than the bubble point, use CCT to calculate the compressibility. Since this routine requires TR in Z, PR in Y, and P in X, which are the stack contents after calling the pressure input routine, set up the stack by executing P.

XEQ ALPHA P ALPHA	*P=?*	
R/S	*2.0000 03*	P (PSI)
XEQ ALPHA CCT ALPHA	*14.532 -06*	CT (1/PSI)

Now use P again to input and store the 1000 PSI pressure from Example 1. Since there is oil present, and this pressure is less than the bubble point, use CCTb to calculate the compressibility.

XEQ ALPHA P ALPHA	*P=?*	
1000 R/S	*1.0000 03*	P (PSI)

After executing P, the stack is set up for CCTb.

XEQ ALPHA CCTb ALPHA	*367.66 -06*	CTb (1/PSI)
R↓	*255.42 00*	RSb (SCF/BBL)
FIX 4	*255.4189*	

Note that the calculation at 1000 PSI was based on no free gas below the bubble point (%SO = 65, %SW = 35 from Example 1). This situation is not physically possible, of course, unless RSI = 0.

```
STD P=15.025E0 PSI
P=1.0000E3 PSI
```

GENERAL PURPOSE, INPUT, AND I/O SUBROUTINES FOR PROGRAMMERS

The HP-41 Petroleum Fluids Pac was designed so that all of the programs are subroutinable. In other words, each program was built out of a series of subroutines that are available for you to use in creating your own programs. As you may have seen, each section of the manual describing the individual programs discussed the calculation subroutine(s) available for that particular program. This section of the manual will discuss the structure of the programs in the Pac, describe the general purpose, input, and I/O subroutines, and illustrate how to use these routines in your own programs.

Naming Conventions Used in the Pac

Each variable used in the Pac is given a name that corresponds to the petroleum industry symbol for that variable. For example, the industry symbol for gas formation volume factor is B_g, and the corresponding Pac variable name is BG. For oil property variables, the name refers to the property above the bubble point. Properties at or below the bubble point have a "BP" or a "B," respectively, on the end of this name.

The name of each program is the name of the variable being calculated by the program. For the above example, the program that calculates gas formation volume factor is called BG. Oil property programs have the name of the variable above the bubble point.

The calculation subroutines are named by placing a "C" before the name of the variable you want to calculate. The calculation subroutine for the BG program is therefore called CBG.

The input subroutines have names corresponding to the variables that are being input. For example, the subroutine to input standard temperature (STD T) and standard pressure (STD P) is called STDTP. Two exceptions to this convention occur when the input subroutine would have the same name as that of a program. The exceptions are resolved by placing an "I" before the name of the input subroutine. TcPc is the name of the *Pseudocritical Temperature and Pressure From Gas Gravity* program, so the subroutine to input Tc and Pc is called ITcPc (*Input Tc and Pc*). Similarly, RS is the *Gas-Oil Ratio* program, so the subroutine to input RS is called IRS (*Input RS*).

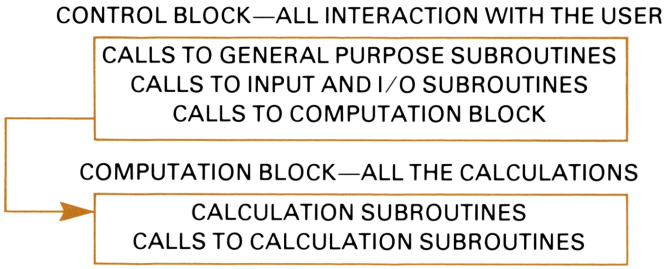
The general purpose and I/O subroutines have names corresponding to their function.

Block Structure

All the programs in the Pac are based on a two-block structure. The first block is the control block. This block controls all the interaction between

the user and the program by calling general purpose, input, and I/O subroutines. In addition, the control block controls which calculations are performed by the program through calls to the second block, which is the computation block.

The computation block is where all the calculations for the program are performed. This block consists of calculation subroutines and calls to calculation subroutines. However, the computation block never calls the control block. This structure is illustrated below.



Since the individual calculation subroutines are specific to a program, instructions for their use are included in the sections for each individual program. Additional details about all the calculation subroutines are in Appendix B. There are, however, some points you should be aware of when using calculation subroutines.

1. Calculation subroutines are named as described previously. Some programs may have more than one calculation subroutine. Oil calculations, for example, have subroutines for conditions above and below the bubble point. Calculation subroutine names below the bubble point are the name above the bubble point followed by a "b". Not all variables have calculation subroutines. Be sure to consult the individual program sections of the manual and Appendix A to get all the information you need to use the calculation subroutines.
2. Calculation subroutines expect certain variables in the registers, the stack, or both. The register contents are used by the subroutine, but the **registers used for input variables (10-44) are never altered by the subroutine.** Some registers (00-02, 05-07) are designated by the Pac as scratch registers. Only these registers and the stack are altered by the subroutine.
3. All values calculated by a calculation subroutine are returned to the calling program **in the stack only**. When a calculated value is to be stored in a register (such as GAS G, calculated by the *Gas Properties From Composition* program), it is stored by the control block, not the computation block.

4. All inputs to a calculation subroutine, whether in registers or in the stack, must be in **Pac English default units**. All values returned in the stack by a calculation subroutine will be in **Pac English default units**. (Since the default units for each value returned in the stack are known, the ALPHA register will not contain the units for those values. This is illustrated in Example 2 of the *Water Viscosity* program.)
5. The status of certain flags is tested by calculation subroutines, but the **flag status is never altered by the subroutine**.
6. Normally, subroutines in the Pac are called with an XROM, which is a subroutine call to or within an e**X**ternal **ROM**. However, all calculation subroutines are called with an XEQ*. An XEQ always searches for the subroutine in 41 program memory before looking at any modules or peripherals. This allows you to calculate a particular variable using a different technique than the one used in the Pac.

   For example, the dead oil viscosity calculation subroutine of the *Oil Viscosity* program is based on the correlation developed by Beggs and Robinson. The calculation subroutine is called CUOd (*Calculate UO for Dead Oil*). If you prefer the correlation developed by Beal, you can write a program starting with LBL CUOd that calculates dead oil viscosity using that correlation. Then, when you run the UO program, the subroutine you have placed in the 41 program memory will be executed instead of the one in the Pac.

The General Purpose Subroutines

The general purpose subroutines simplify several often-done operations: printing a title, checking that enough data registers exist to run the program, and asking the user a question.

> **Note:** Example program listings in the rest of this section are shown with XROMs, as if the programs were entered with the Petroleum Fluids Application Module plugged into a port of the HP-41. For further information on XROMs, refer to Section 11 of the *HP-41C/41CV Operating Manual: A Guide for the Experienced User.*
>
> Also, the number of subroutine levels used by each subroutine includes one level used by the calling program to call the subroutine. This means that a subroutine needing 6 subroutine levels could be called **as** a subroutine, but not **from** a subroutine.

*The machine language functions CZ and CCR are called with an XROM.

TITLE *(Print a Title)*

The control block of every program in this Pac begins by calling the TITLE subroutine. This subroutine prints a double-wide title for the program if an optional printer is plugged into the 41, and checks to see if the size is set correctly for the program. (The routine also sets flag 08—see INK and OUTK).

The routine expects the program title (up to 12 characters) in the ALPHA register, and the minimum number of data registers needed by the program in X. The call to TITLE should be followed with the instructions "FC?C 25 PROMPT". Upon return, flag 25 will be clear if the size is not set correctly, and the program will halt with the following message in the display:

***SIZE*>=nnn**

where nnn is the number of registers in X when TITLE was called. Be aware that if TITLE is called from a subroutine, the [SIZE] function will destroy the subroutine return stack. TITLE uses one subroutine level.

For example, the *Z Factor* program uses TITLE to put up the "Z FACTOR" title and to check for the existence of 18 data registers.

```
02 "Z FACTOR"
03 18
04 XROM "TITLE"
05 FC?C 25
06 PROMPT
```

Program title
Number of registers needed
Check for proper size

Y/N? *(Ask a Yes/No Question)*

It is frequently desirable to ask the user a question with two possible answers, posed in a "yes or no" context. The common way to remember the user's response is by using a flag. The flag is set for a "yes" response, and cleared for a "no" response. The Y/N? subroutine aids in asking such questions.

The routine expects a six-character prompt in the ALPHA register, and the number of the flag to be set or cleared in X. The following things are done by the routine:

1. The characters "? Y/N:" are appended to the end of the prompt in the ALPHA register. Then an additional character is appended after the colon, depending on the current status of the flag. If the flag is set (i.e., the prompt is currently true), the extra character will be a "Y". If the flag is clear (i.e., the prompt is currently false), the extra character will be a "N".
2. ALPHA mode is turned on so the user can respond with Y or N, and is turned off before returning to the point of call.

3. The flag specified in X is set or cleared depending on the user's response. If the response was anything but the character "Y", a negative response is assumed. If there was no response, the current status of the flag is retained.
4. If there was a response, the routine prints the results of the question (if an optional printer is plugged into the 41).

Y/N? uses 1 subroutine level.

The flags used by a program or subroutine are listed in Appendix A, and their meanings are listed in Appendix B.

The *Gas Properties From Composition* program illustrates the use of Y/N?. The program asks whether or not registers are to be cleared. Flag 07 is used to retain the decision.

```
12 "CLEAR"
13 7
14 XROM "Y/N?"
15 26.044
16 FC? 07
17 GTO 01
18 0
19◆LBL 00
20 STO IND Y
21 ISG Y
22 GTO 00
23◆LBL 01
```

Question to be asked
Flag number

If answer was no, don't clear registers

If answer was yes, do clear registers

Input Subroutines

Input subroutines simplify the process of prompting the user for the different input variables used in the Pac. Like calculation subroutines, input subroutines use the stack and the scratch registers listed in Appendix A. All values returned in the stack by an input subroutine will be in **Pac English default units**. (Since the default units for each value returned in the stack are known, the ALPHA register will not contain the units for those values.) The user will be prompted in either English or SI units, depending on the status of flag 09.

The input subroutines are described below in the order that they appear in the catalog ([■] [CATALOG] 2).

COMP *(Input Gas Composition)*

This subroutine asks the user for mole percentages of 19 different constituents of a natural gas (%N2, %CO2, %H2S, %METH, %ETH, %PROP, %IBUT, %N-BUT, %IPEN, %N-PEN, %N-HEX, %N-HEP, %N-OCT, %N-NON, %N-DEC, %O2, %H2, %He, and %H2O), and stores them in registers 26-44. The total of the mole percentages (%TOT) is in X when the routine returns.

This routine allows the user to skip the remainder of the input list by pressing [■] [E] (assuming there is nothing assigned to that key). This is illustrated in Example 1 of the *Gas Properties From Composition* program. For this option to work, the 41 must be in USER mode. The program calling COMP should turn on USER mode by setting flag 27 ([■] [SF] 27).

The calling sequence for this routine requires three program steps: a text name (up to six characters), the call to the subroutine, and a global label that matches the text name. You should select a label (and therefore a name) that does not conflict with any other label that might be in the calculator. The calling sequence used by the *Gas Properties From Composition* program looks like this:

```
24 "W0"
25 XROM "COMP"
26♦LBL "W0"
```

Name

Global label that matches name

Be aware that if COMP is called from a subroutine, the [■] [E] option will destroy the subroutine return stack. COMP uses 3 subroutine levels.

SOUR *(Input Sour Gases)*

This subroutine asks the user for mole percentages of nitrogen, carbon dioxide, and hydrogen sulfide. %N2 is stored in register 26, and %CO2 is stored in register 27. %H2S is stored in register 28, and is in X when the routine returns. SOUR uses 2 subroutine levels.

ITcPc *(Input Tc and Pc)*

This subroutine asks the user for critical or pseudocritical temperature in either R or K. Tc in R is stored in register 10. Then the routine asks the user for critical or pseudocritical pressure in either PSI or KPA. Pc in PSI is stored in register 11, and is in X when the routine returns. ITcPc uses 3 subroutine levels.

STDTP *(Input STD T and STD P)*

This subroutine asks the user for standard temperature in either F or C. STD T in F is stored in register 22. Then the routine asks for standard pressure in either PSI or KPA. STD P in PSI is stored in register 23, and is in X when the routine returns. STDTP uses 4 subroutine levels.

If the previously stored value of STD T is 0 F (i.e., if the registers were cleared) or ALPHA data, the routine will automatically replace that value with 60 F before the ***STD T=?*** prompt appears. If the previously stored value of STD P is 0 PSI or ALPHA data, the routine will automatically replace that value with 14.65 PSI before the ***STD P=?*** prompt appears. There is further discussion of standard temperatures and pressures in the *Gas Formation Volume Factor* program.

SEPTP *(Input SEP T and SEP P)*

This subroutine asks the user for separator temperature in either F or C. SEP T in F is stored in register 24. Then the routine asks for separator pressure in either PSI or KPA. SEP P in PSI is stored in register 25, and is in X when the routine returns. SEPTP uses 4 subroutine levels.

SEP T and SEP P are used by the CGS subroutine to correct the gas gravity for separator conditions. If you do not know these values, you can eliminate the effect of the correction by using 0 F for SEP T (with any positive, nonzero pressure) or 114.7 PSI for SEP P.

Normally, separator pressures are given in gauge pressure units. Consistent with all pressures in the Pac, the separator pressures keyed in must be in absolute pressure units.

T *(Input T)*

This subroutine asks the user for temperature in either F or C. T in F is stored in register 16, and is in X when the routine returns. T uses 3 subroutine levels.

P *(Input P)*

This subroutine asks the user for pressure in either PSI or KPA. P in PSI is stored in register 17, and is in X when the routine returns. Then the routine will compute TR and PR and return with TR in Z, PR in Y, and P in X. If registers 10 (Tc) or 11 (Pc) contain values that would cause the computation of TR or PR to give an error (such as zero or ALPHA data), TR and PR will not be computed, but the value of P will still be returned to X. P uses 4 subroutine levels.

GASG *(Input GAS G or MW)*

This subroutine asks the user for gas gravity. If the user enters a number and presses R/S, GAS G is stored in register 15. If the user presses R/S with no data input, the routine asks for molecular weight. If MW is input, it is converted to GAS G and stored in register 15. GAS G

is in X when the routine returns. If register 15 contained ALPHA data, and there was no data input, the routine will automatically replace that value with 1 before returning to the calling program. GASG uses 3 subroutine levels.

OILG *(Input OIL G)*

This subroutine asks the user for oil gravity in either API or KG/M3. OIL G in API is stored in register 12, and is in X when the routine returns. OILG uses 3 subroutine levels.

IRS *(Input RS)*

This subroutine asks the user for gas-oil ratio in either SCF/BBL or SCM/M3. RS in SCF/BBL is stored in register 13, and is in X when the routine returns. IRS uses 3 subroutine levels.

RSI *(Input RSI)*

This subroutine asks the user for initial gas-oil ratio in either SCF/BBL or SCM/M3. RSI in SCF/BBL is stored in register 13, and is in X when the routine returns. RSI uses 3 subroutine levels.

Notice that RS and RSI are both stored in the same register.

%NACL *(Input %NACL or PPM)*

This subroutine asks the user for weight percent sodium chloride. If the user enters a number and presses R/S, %NACL is stored in register 19. If the user presses R/S with no data input, the routine asks for parts per million. If PPM is input, it is converted to %NACL and stored in register 19. %NACL is in X when the routine returns. If register 19 contained ALPHA data, and there was no data input, the routine will automatically replace that value with 1 before returning to the calling program. %NACL uses 3 subroutine levels.

Notice that PPM is a variable name, and not a unit. PPM cannot be put in a unit equation for use with CON or INCON.

%POR *(Input %POR)*

This subroutine asks the user for percent porosity. %POR is stored in register 18, and is in X when the routine returns. %POR uses 2 subroutine levels.

I/O Subroutines

I/O subroutines simplify the operations needed to prompt for, store, display, and output variables and their units, both with and without an optional printer. In fact, as you will see, the input routines just described were created by calling the I/O routines.

Four of the I/O subroutines involve the use of units. In the discussions of these routines (INU, INK, OUTU, OUTK) it is assumed that you have read and understood *The Petroleum Engineering Unit Management System.*

IN *(Input)*

This subroutine is used to input and store variables that have no units associated with them (e.g., GAS G, %METH, etc.). The routine prompts the user for data input. If the user enters a number and presses R/S, flag 22 is set, the new value is stored, and, if an optional printer is present, the value is printed. If the user presses R/S with no data input, flag 22 is cleared, and the original value of the requested variable is retained. Upon return, the variable will be in X. When the prompt appears for that input, the user can examine the current value of that variable by pressing ←.

IN requires a six-character variable name in the ALPHA register, and a pointer in register 00 that tells where (which data register) the value is to be stored. The value of the pointer must be one less than the number of the register where you want the variable to be stored. For example, to store an input in register 14, register 00 must contain 13 before calling the routine. IN uses 1 subroutine level.

The example below from the *Total Isothermal Compressibility* program shows how %SO and %SW are stored in registers 20 and 21 using IN. Notice that once the pointer is stored in register 00, it is automatically incremented by the subroutine. This allows the calling program to store successive inputs in sequentially numbered registers without resetting the pointer.

```
21 19
22 STO 00
23 "%SO"
24 XROM "IN"
25 "%SW"
26 XROM "IN"
```

Pointer to register 20

Name

%SW will be stored in register 21

INU *(Input with Units)*

This subroutine is used to input and store variables that have units associated with them (e.g., STD T, OIL G, etc.). The routine prompts the user for data input. When the prompt appears, either English or SI default units will be in the ALPHA register, depending on the status of flag 09. If the user enters a number and presses R/S, flag 22 is set, the new value is converted to English default units and stored, and, if an optional printer is present, the value and its units are printed. If the user presses R/S with no data input, flag 22 is cleared, and the original value of the requested variable is retained.

Similarly, if the user presses ALPHA, enters new units, and presses R/S, flag 23 is set, the value is converted to English default units and stored, and the value and its units are printed. If the value could not be

converted to English default units (for example, if the prompt was for pressure and the units supplied by the user were for viscosity), a prompt with the erroneous units will be displayed until a correct set of units have been input. If the user presses R/S with no ALPHA input, flag 23 is cleared, and the original value of the requested variable is retained.

The value in X will also be converted to English default units, stored, and printed if both a number was input and the units were changed. The original value of the requested variable will be retained (but not printed) if the user presses R/S with no data or ALPHA input.

Before calling INU, a six-character variable name must be in ALPHA, and a pointer must be in register 00, as described for IN. Also the English default units for the variable must be stored in registers 01 and 02. The first six characters of the units must be stored in register 01, and the last six characters of the units must be stored in register 02. The SI default units for the variable must be stored in Y and Z (the first six characters in Y, and the last six characters in Z).

Upon return, the value of the input variable in English default units will be in X, and the units that were entered by the user will be in Y and Z (the first six characters will be in Y, and the last six characters will be in Z). This is illustrated in Example 3 of the *Oil Viscosity* program.

Upon return, the English default units will still be in registers 01 and 02. This means that to input successive values with the same units, you only need to place the SI default units in Y and Z prior to calling INU. Also, the pointer in register 00 will be automatically incremented, as it was for IN. INU uses 2 subroutine levels.

The oil gravity input routine (OILG) uses INU to input OIL G in either API or KG/M3 and store it in register 12.

```
160 11
161 STO 00
162 "API"
163 ASTO 01
164 CLA
165 ASTO 02
166 ASTO Z
167 "KG/M3"
168 ASTO Y
169 "OIL G"
170 XROM "INU"
```

Pointer to register 12

Store English default units in registers 01 and 02

Store SI default units in Y and Z

Name

Notice that one step is saved by using the sequence "CLA ASTO 02 ASTO Z KG/M3 ASTO Y" instead of "CLA ASTO 02 KG/M3 ASTO Y CLA ASTO Z".

INK *(Input with Known Units)*

This subroutine is also used to input and store variables that have units associated with them. As just described, INU takes either English default units from registers 01 and 02, or SI default units from Y and Z, depending on the status of flag 09. INK does the same thing **on the first pass through a program only**. On subsequent passes through a program, INK always takes units from Y and Z regardless of the status of flag 09.

The advantage of using INK is that the calling program can save the units that the user entered. Because the units were saved, the user will not have to reenter the units every time the prompt appears for that variable. This is particularly useful when several inputs may have the same units. The disadvantages of using INK are increased program length to save the user's units, and an additional two data registers required per variable to store the units.

On the first pass through a program, it is not known what units the user wants. INK will use either English or SI default units, depending on the status of flag 09. When the routine returns, the variable in English default units will be in X, and the units that were entered by the user will be in Y and Z (the first six characters in Y, and the last six characters in Z).

The calling program must save these units. When this prompt is reached again, the program must place the saved units in Y and Z (the first six characters in Y, and the last six characters in Z). Now INK can prompt in the user's units instead of the default units.

The way INK distinguishes between the first pass through a program and subsequent passes is by flag 08. If flag 08 is set, it is the first pass. If flag 08 is clear, it is another pass. Flag 08 is set by the TITLE routine, but must be cleared by the program using NK.

Before calling INK, a six-character variable name must be in ALPHA, and a pointer must be in register 00, as described for IN. On every pass through the program, English default units must be in registers 01 and 02 as described for INU. On the first pass through the program, SI default units must be in Y and Z, and on subsequent passes through the program, the user's known input units must be in Y and Z. INK uses 2 subroutine levels.

The following example illustrates the use of INK to input "PRESS" and store it in register 11. The example assumes that flag 08 was set by calling TITLE at the beginning of the program. The units that were entered by the user will be saved in registers 03 and 04, although almost any two registers could have been used (see the *Notes About the I/O Subroutines*).

Notice that early in the example, SI default units are saved in registers 03 and 04 so that on the first pass, the user will be prompted with either English or SI default units. Also notice that flag 08 is cleared after the first pass through the example.

```
07 "KPA"
08 ASTO 03
09 CLA
10 ASTO 04
11♦LBL 00
12 "PSI"
13 ASTO 01
14 CLA
15 ASTO 02
16 RCL 04
17 RCL 03
18 10
19 STO 00
20 "PRESS"
21 XROM "INK"
22 RDN
23 STO 03
24 RDN
25 STO 04
26 CF 08
27 GTO 00
```

Store SI default units in registers 03 and 04 on 1st pass only

Store English default units in registers 01 and 02 on all passes

Last six chars. of units put in Z, 1st six in Y (these are SI units on 1st pass)
Pointer to register 11

Name

Units entered by user returned to Y and Z
Store 1st six chars. in register 03, last six in register 04

No longer 1st pass
Loop for next input

OUT *(Output)*

This subroutine is used to output variables that have no units associated with them (e.g., GAS GS, BO, etc.). The routine requires a six-character variable name in ALPHA, and the value of the output variable in X. OUT uses 2 subroutine levels.

As an example, GAS GS is output using the OUT subroutine.

```
95 XEQ "CGS"
96 "GAS GS"
97 XROM "OUT"
```

Calculate GAS GS; value returned to X
Name

OUTU *(Output with Units)*

This subroutine is used to output variables that have units associated with them (e.g., Tc, RSb, etc.). Before calling OUTU, a six-character variable name must be in ALPHA, and the value of the output variable in English default units must be in X. The English default units must be stored in registers 01 and 02, and the SI default units must be stored in Y and Z, as described for INU.

Upon return, the value of the output variable in English default units will be in X, and the units that were entered by the user will be in Y and Z, as described for INU. Unlike INU, however, the English default units will **not** be in registers 01 and 02. Therefore, to output successive values with the same units, you must place the English default units in registers 01 and 02 as well as the SI default units in Y and Z prior to calling OUTU. OUTU uses 2 subroutine levels.

Here is an example using OUTU to output BG.

```
40 XROM "CBG"
41 "FT3/SCF"
42 ASTO 01
43 ASHF
44 ASTO 02
45 "M3/SCM"
46 ASTO Y
47 CLA
48 ASTO Z
49 "BG"
50 XROM "OUTU"
```

Calculate BG; value returned to X

Store English default units in registers 01 and 02

Store SI default units in Y and Z

Name

OUTK *(Output with Known Units)*

This subroutine is also used to output variables that have units associated with them. OUTK bears the same relationship to OUTU that INK does to INU. On the first pass through a program, OUTK takes either English default units from registers 01 and 02, or SI default units from Y and Z, depending on the status of flag 09. On subsequent passes through a program, the units are always taken from Y and Z regardless of the status of flag 09.

The same techniques for using INK apply to OUTK, including the use of flag 08. The programs in the Pac use OUTK to save the user's output units, and the units are saved in registers 03 and 04. (Since only two registers are used, only one set of output units can be saved. This is why only the units for the primary output variable of a Pac program are saved, and why the output units for Tc, Pc, and CWA are not saved by the *Pseudocritical Temperature and Pressure From Gas Gravity* or *Gas Properties From Composition* programs.)

Before calling OUTK, a six-character variable name must be in ALPHA, and the value of the output variable in English default units must be in X. On every pass through the program, the English default units must be in registers 01 and 02. On the first pass through the program, SI default units for the output variable must be in Y and Z, and on subsequent passes through the program, the user's known output units must be in Y and Z.

Upon return, the value of the output variable in English default units will be in X, and the units that were entered by the user will be in Y and Z, as described for INK. As described for OUTU, the English default units will **not** be in registers 01 and 02. OUTK uses 2 subroutine levels.

The following example illustrates the use of this routine to output "PRESS". As in the example for INK, it is assumed that flag 08 was set earlier in the program.

```
34 "KPA"
35 ASTO 03
36 CLA
37 ASTO 04
38♦LBL 01
39 XEQ "CPRESS"
40 "PSI"
41 ASTO 01
42 CLA
43 ASTO 02
44 RCL 04
45 RCL 03
46 RCL Z
47 "PRESS"
48 XROM "OUTK"
49 RDN
50 STO 03
51 RDN
52 STO 04
53 CF 08
54 GTO 01
```

Store SI default units in registers 03 and 04 on 1st pass only

Calculate PRESS; value returned to X

Store English default units in registers 01 and 02 on all passes

Last six chars. of units put in Z, 1st six in Y (these are SI units on 1st pass)
Get value back in X

Name

Units entered by user returned to Y and Z
Store 1st six chars. in register 03, last six in register 04

No longer 1st pass
Loop for next output

Notes About the I/O Subroutines

If you want to save program space when writing your own programs using INU, INK, OUTU, and OUTK, you can eliminate placing the SI default units in Y and Z. If you do this, however, flag 09 **must** be clear for the I/O routines to function properly, and you will lose the benefit of the SI option for those programs.

The I/O subroutines use registers 00-02 and 05. The Pac uses registers 03 and 04 for known output units (see OUTK). Because of this, you should not store any input or output variables in registers 00-05.

Remember that whenever INU, INK, OUTU, and OUTK return, the number in X will **always** be in English default units, and the units entered by the user will **always** be in Y and Z. (Since the default units for the number in X are known, the ALPHA register will not contain those units.)

Tables 6 and 7 summarize how to call the I/O subroutines and what is returned by them.

Table 6: How to Call I/O Subroutines

	IN	INU	INK pass 1	INK pass > 1	OUT	OUTU	OUTK pass 1	OUTK pass > 1
Z	NR	SI default units	SI default units	User's known units	NR	SI default units	SI default units	User's known units
Y								
X	NR	NR	NR	NR	value	value (Eng. units)	value (Eng. units)	value (Eng. units)
R00	pointer	pointer	pointer	pointer	NR	NR	NR	NR
R01	NR	English default units	English default units	English default units	NR	English default units	English default units	English default units
R02								
F08	NR	NR	set	clear	NR	NR	NR	NR
ALPHA	Name	Name	Name	Name	Name	Name	Name	Name

NR = nothing required

Table 7: What Is Returned by I/O Subroutines

	IN	INU	INK pass 1	INK pass > 1	OUT	OUTU	OUTK pass 1	OUTK pass > 1
Z	null data	Units entered by user	Units entered by user	Units entered by user	null data	Units entered by user	Units entered by user	Units entered by user
Y								
X	value	value (Eng. units)	value (Eng. units)	value (Eng. units)	value	value (Eng. units)	value (Eng. units)	value (Eng. units)
R00	VC + 1	VC + 1	VC + 1	VC + 1	NU	NU	NU	NU
R01	VC	English default units	English default units	English default units	NU	NU	NU	NU
R02								
F08	VC	VC	VC	VC	VC	VC	VC	VC
ALPHA	NU	NU	NU	NU	NU	NU	NU	NU

VC = value when the subroutine is called
NU = not usable

Number of Characters Allowed for English Default Units

The English default units for your variables must be no more than 11 characters long. INU, INK, OUTU, and OUTK perform automatic unit conversions by building up a unit equation in the ALPHA register to convert the user's units to English default units (see the *Using* CON *and* INCON section of *The Petroleum Engineering Unit Management System*). When the unit equation is created, it could have as many as 12 characters entered by the user, a dash, and 11 characters from the English default units—24 characters total, the limit of the ALPHA register. (Units up to 12 characters long are allowed for the user because two data registers holding six characters each are used to store them.) If your English default units are 12 characters long, the units entered by the user must be limited to 11 characters.

Developing a Program Using Pac Subroutines

Now that you understand how these routines function, you can combine them into useful programs. The individual sections on each program, *General Purpose, Input, and I/O Subroutines for Programmers,* and Appendices A and B are used extensively to determine which subroutines to use, what size is needed, what variables must be input, where they are stored, what must be in the stack when the routines are called, what is in the stack when the routines return, which scratch registers are available, and which flags must be considered.

To illustrate the process of combining the Pac subroutines into other programs, the following example will be used. A program will be written to output values of Z, BG, and UG as a function of pressure. In addition to the fluid property variables, the user will input a starting pressure, a pressure increment, and the desired number of increments.

First, let's examine the inputs. Below is a table that summarizes the fluid property variables used by the Z, BG, and UG programs.

	Tc	Pc	STD T	STD P	GAS G	T	P
Z	Y	Y	—	—	—	Y	Y
BG	Y	Y	Y	Y	—	Y	Y
UG	Y	Y	—	—	Y	Y	Y

We can see that the required fluid property variables for the example are Tc, Pc, STD T, STD P, GAS G, T, and P. To input these values will require ITcPc (*Input Tc and Pc*), STDTP (*Input STD T and STD P*), GASG (*Input GAS G or MW*), T (*Input T*), and P (*Input P*).

We also need to input the pressure increment (P INC) and the number of increments (NO INC). Since registers 08 and 09 are never used by any programs in the Pac, we will store P INC in 08 and NO INC in 09. The pressure increment has units, so we will use INU to input and store P INC. Since the number of increments is dimensionless, we will use IN to input and store NO INC.

Next, let's examine the calculation subroutines. We want to calculate Z, BG, and UG. The subroutines needed are CZ (*Calculate Z*), CBG (*Calculate BG*), and CUG (*Calculate UG*). All three calculation subroutines need TR in Y and PR in X when they are called, and return Z, BG in FT3/SCF, and UG in CP, respectively, to X. Scratch registers 00 and 05 are used by CUG, and no flags are used.

Now let's consider the outputs. We will use OUT to output the dimensionless Z factor. OUTU will be used to output BG and UG, since both have units associated with them.

Finally, let's examine the general purpose subroutines. The only one needed is TITLE (*Print a Title*). The highest number register needed for the input variables is 23, used to store STD P. Therefore, size 024 is required. Y/N? (*Ask a Yes/No Question*) is not needed because no flags are tested by any of the calculation subroutines.

The completed program is shown below.

```
01♦LBL "ZBU"
02 "Z BG UG"
03 29
04 XROM "TITLE"
05 FC?C 25
06 PROMPT
07 XROM "ITcPc"
08 XROM "STDTP"
09 XROM "GASG"
10 XROM "T"
11♦LBL 00
12 XROM "P"
13 7
14 STO 00
15 "KPA"
16 ASTO Y
17 CLA
18 ASTO Z
19 "P INC"
20 XROM "INU"
21 "NO INC"
22 XROM "IN"
23 ADV
24♦LBL 01
25 RCL 17
26 "PSI"
27 ASTO 01
28 CLA
29 ASTO 02
30 ASTO Z
31 "KPA"
32 ASTO Y
33 "P"
34 XROM "OUTU"
35 XEQ 02
36 CZ
37 "Z"
38 XROM "OUT"
39 XEQ 02
40 XROM "CBG"
```

Initialize

Input fluid property variables

Starting pressure

Input P INC and store in register 08
English default units for P INC already in registers 01 and 02 after "XROM P"

Input NO INC and store in register 09

Space for readability

Output current P

Calculate TR and PR
Calculate Z

Output Z

Calculate TR and PR
Calculate BG

```
41 "FT3/SCF"
42 ASTO 01
43 ASHF
44 ASTO 02
45 "M3/SCM"
46 ASTO Y
47 CLA
48 ASTO Z
49 "BG"
50 XROM "OUTU"
51 XEQ 02
52 XROM "CUG"
53 "CP"
54 ASTO 01
55 CLA
56 ASTO 02
57 ASTO Z
58 "PA*S"
59 ASTO Y
60 "UG"
61 XROM "OUTU"
62 ADV
63 RCL 08
64 ST+ 17
65 DSE 09
66 GTO 01
67 GTO 00
68*LBL 02
69 RCL 16
70 "F-R"
71 CON
72 RCL 16
73 /
74 RCL 17
75 RCL 11
76 /
77 END
```

Output BG

Calculate TR and PR

Calculate UG

Output UG

Space for readability

Get P INC

New P = old P + P INC

Count down NO INC

If NO INC > 0, loop back and do again

If NO INC = 0, prompt for new starting P

Calculate TR and PR

T

Tc

TR

P

Pc

PR

Try this program for a gas at 300 F with a Tc of 383 R and a Pc of 45.4 ATM. The gas gravity is 0.74 and the standard conditions are 14.65 PSI and 60 F. Start at a pressure of 500 PSI, and generate a table of Z, BG, and UG every 500 PSI. The values that are calculated at the first four pressures are shown below.

```
Z  BG  UG

Tc=383.0000 R
Pc=45.4000 ATM
STD T=60.0000 F
STD P=14.6500 PSI
GAS G=0.7400
T=300.0000 F
P=500.0000 PSI
P INC=500.0000 PSI
NO INC=4.0000

P=500.0000 PSI
Z=0.9734
BG=0.0417 FT3/SCF
UG=0.0153 CP

P=1000.0000 PSI
Z=0.9530
BG=0.0204 FT3/SCF
UG=0.0159 CP

P=1500.0000 PSI
Z=0.9400
BG=0.0134 FT3/SCF
UG=0.0168 CP

P=2000.0000 PSI
Z=0.9350
BG=0.0100 FT3/SCF
UG=0.0178 CP
```

Summary of Labels Used in the Pac

Table 8 lists all the labels in the Pac by category (program or subroutine, and type of subroutine). The order of the entries in each category is the order that they appear in the catalog (CATALOG 2).

Table 8: Summary of Labels Used in the Pac

Program	Subroutines					Unit Mgmt.	Internal Use**
	Calculation		General	Input	I/O		
Z	CCG	CBW	TITLE	COMP	IN	CON	W0
CG	CBG	CUW	Y/N?	SOUR	OUTK	INCON	W1
BG	CUG	CPSAT		ITcPc	OUTU		W2
UG	CTcPc	CCFR		STDTP	OUT		W3
TcPc	CCWA	CRSb		SEPTP	INK		W4
PROP	CGASG	CPBP		T	INU		W5
CO	CTPC	CBTb		P			W6
BO	CHV	CRSW		GASG			W7
UO	CCK	CCT		OILG			W8
CW	CCOb	CCTb		IRS			W9
BW	CCO	CZ		RSI			X0
UW	CGS	CCR		%NACL			X1
CFR	CBOb			%POR			X2
RS	CBO						X3
PBP	CBT						X4
BT	CUOd						X5
RSW	CUOb						X6
CT	CUO						X7
	CCW						X8

**These labels are for internal use by the programs in the Pac.

Z FACTOR

```
 01♦LBL "Z"
"Z FACTOR"  18
XROM "TITLE"  FC?C 25
PROMPT  XROM "ITcPc"
XROM "T"

 09♦LBL 00
XROM "P"  RDN  CZ
FS? 08  ADV  "Z"
XROM "OUT"  ADV  CF 08
GTO 00  END
```

Initialize

Input variables

Input P
Calculate and output Z

Loop back for new P

GAS ISOTHERMAL COMPRESSIBILITY

```
 01♦LBL "CG"
"GAS ISO CMP"  18
XROM "TITLE"  FC?C 25
PROMPT  "1/KPA"  ASTO 03
CLA  ASTO 04
XROM "ITcPc"  XROM "T"

 13♦LBL 00
XROM "P"  RDN  XEQ "CCG"
"CG"  XROM "X2"  ADV
GTO 00

 21♦LBL "CCG"
CCR  RCL 11  /  END
```

Initialize

Store SI default units for CG in registers 03 and 04
Input variables

Input P
Calculate and output CG
Loop back for new P

Calculate CG

GAS FORMATION VOLUME FACTOR

```
 01♦LBL "BG"
"GAS VOL FACT"  24
XROM "TITLE"  FC?C 25
PROMPT  "M3/SCM"
ASTO 03  CLA  ASTO 04
XROM "ITcPc"
XROM "STDTP"  XROM "T"

 14♦LBL 00
XROM "P"  RDN  XEQ "CBG"
"BG"  XROM "X1"  ADV
GTO 00

 22♦LBL "CBG"
CZ  LASTX  RCL 11  *  /
RCL 16  "F-R"  CON  *
RCL 23  *  RCL 22  CON
/  END
```

Initialize

Store SI default units for BG in registers 03 and 04
Input variables

Input P
Calculate and output BG
Loop back for new P

Calculate BG

GAS VISCOSITY

```
 01♦LBL "UG"
"GAS VIS"  18
XROM "TITLE"  FC?C 25
PROMPT  "PA#S"  ASTO 03
CLA  ASTO 04
XROM "ITcPc"
XROM "GASG"  XROM "T"

 14♦LBL 00
XROM "P"  RDN  XEQ "CUG"
"UG"  XROM "X3"  ADV
GTO 00

 22♦LBL "CUG"
CZ  RCL 16  "F-R"  CON
STO 00  *  RCL 17  X<>Y
/  669.8  /  RCL 15
28.964  *  STO 05  *
LASTX  100  /  986
RCL 00  /  +  3.5  +
STO Z  -5  /  2.4  +
Y↑X  *  E↑X  RCL 00
SQRT  3  Y↑X  1 E4  /
RCL 05  50  /  9.4  +  *
RCL 05  19  *  209  +
RCL 00  +  /  *  END
```

Initialize

Store SI default units for UG in registers 03 and 04

Input variables

Input P

Calculate and output UG

Loop back for new P

Calculate UG

PSEUDOCRITICAL TEMPERATURE AND PRESSURE FROM GAS GRAVITY

```
 01♦LBL "TcPc"
"Tc Pc"  29
XROM "TITLE"  FC?C 25
PROMPT  "COND"  5
XROM "Y/N?"

 10♦LBL 00
XROM "GASG"  XROM "SOUR"
XEQ "CTcPc"  FS? 08  ADV
XROM "X0"  ADV  CF 08
GTO 00

 20♦LBL "CTcPc"
RCL 15  100  *  STO 00
LASTX  RCL 26  RCL 27  +
RCL 28  +  -  STO 01
RCL 00  RCL 26  .9672  *
-  RCL 27  1.5195  *  -
RCL 28  1.1765  *  -
X<>Y  /  STO 00  FS? 05
GTO 01  325  RCL 00
12.5  *  -  RCL 00  *
168  +  15  RCL 00  37.5
*  -  RCL 00  *  677  +
GTO 02

 70♦LBL 01
330  RCL 00  71.5  *  -
RCL 00  *  187  +  706
RCL 00  11.1  *  51.7  +
RCL 00  *  -

 89♦LBL 02
RCL 01  ST* Z  *  RCL 26
493  *  +  RCL 27  1071
*  +  RCL 28  1306  *  +
X<>Y  RCL 26  227.3  *
+  RCL 27  547.6  *  +
RCL 28  672.4  *  +
X<>Y  100  ST/ Z  /
STO 00  X<>Y  STO 05
```

Initialize

Flag 05: Condensate fluid or miscellaneous gas

Input GAS G and sour gases
Calculate and output Tc, Pc, CWA, Tc*, and Pc*
Loop back for new GAS G and sour gases

Calculate Tc and Pc

Calculate GAS G_{HC}

Calculate Tc_{HC} and Pc_{HC} for miscellaneous gases

Calculate Tc_{HC} and Pc_{HC} for condensate fluids

Calculate Tc and Pc from Tc_{HC} and Pc_{HC}

```
X<>Y  RCL 27  RCL 28  +
X=0?  GTO 15  XEQ 15
XEQ "CCWA"  RTN

134♦LBL "CCWA"
STO 00  X<>Y  STO 05
RCL 27  RCL 28  +  100
/  RCL X  .9  Y↑X  X<>Y
1.6  Y↑X  -  8  *
RCL 28  100  /  SQRT
LASTX  X↑2  X↑2  -  +
15  *  STO 01  -  RCL 00
*  1  RCL 28  100  /  -
LASTX  *  RCL 01  *
RCL 05  +  /  RCL 05
RCL 01  -  X<>Y  RCL 01

184♦LBL 15
STO L  CLX  RCL 05
RCL 00  RDN  RDN  END
```

If CO_2 or H_2S present, calculate CWA
If not, return

Calculate CWA, Tc*, and Pc*

		T: Tc
Z: Tc*		Z: Pc
Y: Pc*	⟶	Y: Tc*
X: CWA		X: Pc*
		L: CWA

GAS PROPERTIES FROM COMPOSITION

```
 01♦LBL "PROP"
"GAS PROP"  45
XROM "TITLE"  FC?C 25
PROMPT  SF 27  "KJ/SCM"
ASTO 03  CLA  ASTO 04
"CLEAR"  7  XROM "Y/N?"
26.044  FC? 07  GTO 01
0

 19♦LBL 00
STO IND Y  ISG Y  GTO 00

 23♦LBL 01
"W0"  XROM "COMP"

 26♦LBL "W0"
ADV  "%TOT"  XROM "OUT"
XEQ "CGASG"  STO 15
"GAS G"  XROM "OUT"
XEQ "CTPC"  XROM "X0"
XEQ "CHV"  STO 07  RDN
STO 06  RDN  "NHV"
XROM "X4"  RCL 06
"GHVD"  XROM "X4"
RCL 07  "GHVW"
XROM "X4"  ADV  "SP.HTS"
4  XROM "Y/N?"  FC? 04
GTO 01  SF 08  "KJ/KG*K"
ASTO 03  ASHF  ASTO 04

 60♦LBL 02
XROM "T"  XEQ "CCK"
STO 07  RDN  STO 06  RDN
"CP"  XROM "X5"  RCL 06
"CV"  XROM "X5"  RCL 07
"K"  XROM "OUT"  ADV
GTO 02

 77♦LBL "COMP"
ASTO 06  XROM "SOUR"
"%METH"  XROM "IN"
"%ETH"  XROM "IN"
```

Initialize
Store SI default units for NHV, GHVD, and GHVW in registers 03 and 04
Flag 07: Clear constituent registers or leave unchanged

Loop to clear registers

LBL W0 is where "COMP" will return to

Output %TOT

Calculate and output GAS G, Tc, Pc, CWA, Tc*, Pc*, NHV, GHVD, and GHVW

Flag 04: Calculate CP, CV, and K or loop back to input new composition
If flag 04 set, store SI default units for CP and CV in registers 03 and 04

If flag 04 set, input T
Calculate and output CP, CV, and K

Loop back for new T

Save return "address" (W0) in register 06

```
"%PROP"  XROM "IN"
"%IBUT"  XROM "IN"
"%N-BUT"  XROM "IN"
"%IPEN"  XROM "IN"
"%N-PEN"  XROM "IN"
"%N-HEX"  XROM "IN"
"%N-HEP"  XROM "IN"
"%N-OCT"  XROM "IN"
"%N-NON"  XROM "IN"
"%N-DEC"  XROM "IN"
"%O2"  XROM "IN"  "%H2"
XROM "IN"  "%He"
XROM "IN"  "%H2O"
XROM "IN"  GTO 03

113♦LBL "SOUR"
25  STO 00  "%N2"
XROM "IN"  "%CO2"
XROM "IN"  "%H2S"

121♦LBL "IN"
AOFF  ASTO 05  CF 22
ISG 00  CLD  RCL IND 00
"⊢=?"  CF 21  AVIEW  CLA
FS? 55  SF 21  STOP
AOFF  CLA  ASTO Y
ASTO Z  FS? 22  GTO 05
CLX  RCL IND 00  CF 21
FS? 55  SF 21  RTN

147♦LBL e
FS? 22  XEQ 05

150♦LBL 03
26.044  ENTER↑  0

154♦LBL 04
RCL IND Y  +  ISG Y
GTO 04  GTO IND 06

160♦LBL 05
AOFF  STO IND 00  CF 21
FC? 55  RTN  CLA
ARCL 05  "⊢="  ARCL X
```

Input composition

Go calculate %TOT

Input sour gases

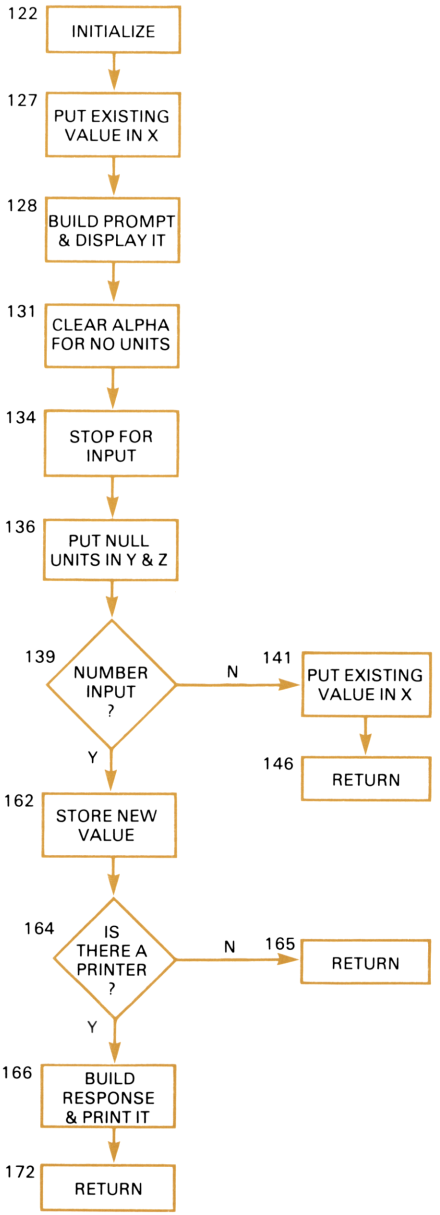
"IN" I/O subroutine—see flowchart

E option
If data input, store it

Calculate %TOT
Return to calling program

More of "IN"

IN



Note: Small numbers are the line numbers in the program listing.

GAS PROPERTIES FROM COMPOSITION (cont.)

```
SF 21  PRA  RTN

173↓LBL "CGASG"
25  STO 00  CLST  .9672
XROM "W3"  1.5195
XROM "W3"  1.1765
XROM "W3"  .5539
XROM "W3"  1.0382
XROM "W3"  1.5225
XROM "W3"  2.0068
XROM "W3"  2.0068
XROM "W3"  2.4911
XROM "W3"  2.4911
XROM "W3"  2.9753
XROM "W3"  3.4596
XROM "W3"  3.9439
XROM "W3"  4.4282
XROM "W3"  4.9125
XROM "W3"  1.1048
XROM "W3"  .0696
XROM "W3"  .138
XROM "W3"  .622
XROM "W3"  100  /  RTN

218↓LBL "CTPC"
25  STO 00  CLST  227.3
ENTER↑  493  XROM "W1"
547.6  ENTER↑  1071
XROM "W1"  672.4  ENTER↑
1306  XROM "W1"  343.04
ENTER↑  667.8  XROM "W1"
549.76  ENTER↑  707.8
XROM "W1"  665.68
ENTER↑  616.3  XROM "W1"
734.65  ENTER↑  529.1
XROM "W1"  765.32
ENTER↑  550.7  XROM "W1"
828.77  ENTER↑  490.4
XROM "W1"  845.4  ENTER↑
488.6  XROM "W1"  913.4
```

Calculate GAS G

Calculate Tc and Pc

GAS PROPERTIES FROM COMPOSITION (cont.)

```
ENTER↑  436.9  XROM "W1"
972.5  ENTER↑  396.8
XROM "W1"  1023.89
ENTER↑  360.6  XROM "W1"
1070.35  ENTER↑  332
XROM "W1"  1111.8
ENTER↑  304  XROM "W1"
278.6  ENTER↑  736.9
XROM "W1"  59.9  ENTER↑
188.1  XROM "W1"  9.5
ENTER↑  33.2  XROM "W1"
1165.3  ENTER↑  3208
XROM "W1"  100  ST/ Z  /
RCL 27  RCL 28  +  X≠0?
GTO 06  +  R↑  RCL Y
RTN

310*LBL 06
RDN  XEQ "CCWA"  RTN

314*LBL "W1"
ISG 00  CLD  STO L  CLX
RCL IND 00  ST* L  ST* Y
RDN  ST+ Z  RDN  LASTX
+  RTN

328*LBL "CHV"
27  STO 00  CLST  588
ENTER↑  637  XROM "W1"
909.1  ENTER↑  1009.7
XROM "W1"  1617.8
ENTER↑  1768.8
XROM "W1"  2316.1
ENTER↑  2517.4
XROM "W1"  3001.1
ENTER↑  3252.7
XROM "W1"  3010.4
ENTER↑  3262.1
XROM "W1"  3698.3
ENTER↑  4000.3
XROM "W1"  3707.5
ENTER↑  4009.5
XROM "W1"  4403.7
```

Calculate Tc and Pc

If CO2 or H2S present, calculate CWA
If not, return

Calculate CWA

T: $\Sigma y_i F_i$
Z: $\Sigma y_i G_i$
Y: F_{i+1} → Y: $\Sigma y_i F_i + y_{i+1} F_{i+1}$
X: G_{i+1} → X: $\Sigma y_i G_i + y_{i+1} G_{i+1}$

F = Tc or NHV, G = Pc or GHVD

No heating values for N2 and CO2

Calculate NHV and GHVD

```
ENTER↑  4756.1
XROM "W1"  5100.2
ENTER↑  5502.9
XROM "W1"  5796.7
ENTER↑  6249.7
XROM "W1"  6493.3
ENTER↑  6996.6
XROM "W1"  7188.6
ENTER↑  7742.3
XROM "W1"  ISG 00  CLD
274  ENTER↑  324
XROM "W1"  100  ST/ Z  /
RCL X  .9826  *  RTN

397↑LBL "CCK"
25  STO 00  RCL 16
"F-R"  CON  XEQ 07  5.87
ENTER↑  .00556
XROM "W2"  7.16  ENTER↑
.00183  XROM "W2"  5.343
ENTER↑  .006032
XROM "W2"  3.782  ENTER↑
.01647  XROM "W2"  3.324
ENTER↑  .02662
XROM "W2"  3.857  ENTER↑
.03593  XROM "W2"  5.104
ENTER↑  .03397
XROM "W2"  4.667  ENTER↑
.04417  XROM "W2"  6.259
ENTER↑  .04197
XROM "W2"  6.972  ENTER↑
.05065  XROM "W2"  8.027
ENTER↑  .05883
XROM "W2"  9.13  ENTER↑
.006693  XROM "W2"
10.29  ENTER↑  .07496
XROM "W2"  11.4  ENTER↑
.08303  XROM "W2"  6.545
```

No heating values for O2, He, and H2O

Calculate GHVW

Special case for N2

Calculate CP

GAS PROPERTIES FROM COMPOSITION (cont.)

```
ENTER↑  8859 E-7
XROM "W2"  6.551  ENTER↑
5914 E-7  XROM "W2"
4.97  ENTER↑  0
XROM "W2"  7.587  ENTER↑
8195 E-7  XROM "W2"  100
/  STO Y  1.987  -
RCL 15  28.964  *  ST/ Z
/  RCL Y  RCL Y  /  RTN

490↑LBL 07
0  RCL Y  LN  .09017  *
6.391  +  GTO 08

499↑LBL "W2"
R↑  *  LASTX  RDN  +

505↑LBL 08
506↑LBL "W3"
ISG 00  CLD  RCL IND 00
*  +  END
```

Calculate CV
Calculate K

Special case for N2

T: T′
Z: $\Sigma y_i(A_i + B_iT')$ → Z: T′
Y: A_{i+1} → Y: $\Sigma y_i(A_i + B_iT')$
X: B_{i+1} → X: $A_{i+1} + B_{i+1}T'$

Y: Σy_iH_i
X: H_{i+1} → X: $\Sigma y_iH_i + y_{i+1}H_{i+1}$

H = GAS G or A + BT′

OIL ISOTHERMAL COMPRESSIBILITY

```
 01♦LBL "CO"
"OIL ISO CMP"  26
XROM "TITLE"  FC?C 25
PROMPT  "1/KPA"  ASTO 03
CLA  ASTO 04  XROM "W7"

 12♦LBL 00
XROM "P"  RCL 14  X<Y?
GTO 01  RDN  XEQ "CCOb"
STO 06  X<>Y  XROM "X8"
RCL 06  "COb"  GTO 02

 25♦LBL 01
XEQ "CCO"  "CO"

 28♦LBL 02
XROM "X2"  ADV  GTO 00

 32♦LBL "CCOb"
STO 05  RDN  XEQ "CBG"
"FT3-BBL"  CON  X<> 05
XEQ "CBOb"  R↑  X<> 05
R↑  *  LASTX  R↑  *  -
X<>Y  /  RCL 05  X<>Y
RTN

 53♦LBL "CCO"
RCL 12  12.61  *
XEQ "CGS"  1180  *  -
RCL 16  17.2  *  +
RCL 13  5  *  +  1433  -
RCL 17  /  1 E5  /  RTN

 76♦LBL "CGS"
RCL 25  114.7  /  LOG
RCL 24  *  RCL 12  *
5912 E-8  *  1  +
RCL 15  *  RTN

 92♦LBL "W4"
XROM "GASG"

 94♦LBL "W5"
XEQ "CGS"  "GAS GS"
XROM "OUT"  END
```

Initialize
Store SI default units for CO in registers 03 and 04
Input variables, calculate and output GAS GS and PBP

Input P
If P ⩽ PBP, calculate and output RSb and COb

If P > PBP, calculate and output CO

Loop back for new P

Calculate COb

Calculate CO

Calculate GAS GS

Input GAS G

Calculate and output GAS GS

OIL FORMATION VOLUME FACTOR

```
 01♦LBL "BO"
"OIL VOL FACT"  26
XROM "TITLE"  FC?C 25
PROMPT  XROM "W8"
XEQ "CBOb"  STO 07
"BOBP"  GTO 02

 12♦LBL 00
XROM "P"  RCL 14  X<Y?
GTO 01  RDN  XEQ "CBOb"
STO 06  R↑  XROM "X8"
RCL 06  "BOb"  GTO 02

 25♦LBL 01
RCL 07  XEQ "CBO"  "BO"

 29♦LBL 02
FS? 08  ADV  XROM "OUT"
ADV  CF 08  GTO 00

 36♦LBL "CBOb"
XEQ "CRSb"  STO 01  X<>Y
STO 02  RCL 16  60  -
RCL 12  XEQ "CGS"  /  *
30  RCL 12  X>Y?  GTO 04
RCL Z  1751 E-8  ENTER↑
4677 E-7  ENTER↑
-1.811 E-8  GTO 05

 59♦LBL 04
RCL Z  11 E-6  ENTER↑
467 E-6  ENTER↑
1.337 E-9

 66♦LBL 05
R↑  *  R↑  LASTX  *  RDN
+  STO Y  RCL 01  *
RCL 02  X<> T  +  1  +
RCL 01  RDN  RTN

 85♦LBL "CBO"
 86♦LBL "CBT"
XEQ "CCO"  RCL 14
RCL 17  -  *  E↑X  *
END
```

Initialize

Input variables, calculate and output GAS GS, PBP, and BOBP

Input P
If P ⩽ PBP, calculate and output RSb and BOb

If P > PBP, calculate and output BO

Loop back for new P

Calculate BOb or BOBP
Calculate RSb, OIL G/GAS GS

A, B, and C for OIL G ⩽ 30 API

A, B, and C for OIL G > 30 API

Calculate BOb from A, B, C, RSb, and OIL G/GAS GS

Calculate BO or BT

OIL VISCOSITY

```
 01♦LBL "UO"
"OIL VIS"  26
XROM "TITLE"  FC?C 25
PROMPT  "PA*S"  ASTO 03
CLA  ASTO 04
XROM "SEPTP"
XROM "OILG"  XROM "W4"
XROM "T"  XEQ "CUOd"
STO 06  "UOd"  XROM "X3'
ADV  XROM "RSI"
XROM "W9"  RCL 13
RCL 06  XEQ "CUOb"
STO 07  "UOBP"  GTO 02

 28♦LBL 00
XROM "P"  RCL 14  X<Y?
GTO 01  RDN  XEQ "CRSb"
XROM "X8"  RCL 06
XEQ "CUOb"  "UOb"
GTO 02

 40♦LBL 01
RCL 07  XEQ "CUO"  "UO"

 44♦LBL 02
XROM "X3"  ADV  GTO 00

 48♦LBL "CUOd"
3.0324  RCL 12  .02023
*  -  10↑X  RCL 16
-1.163  Y↑X  *  10↑X  1
-  RTN

 63♦LBL "CUOb"
RCL Y  150  +  -.338
Y↑X  5.44  *  Y↑X  X<>Y
100  +  -.515  Y↑X  *
10.715  *  RTN

 81♦LBL "CUO"
RCL 17  -898 E-7  *
11.513  -  E↑X  RCL 17
1.187  Y↑X  *  2.6  *
RCL 17  RCL 14  /  X<>Y
Y↑X  *  END
```

Initialize

Store SI default units for UO in registers 03 and 04

Input variables, calculate and output GAS GS

Input variables, calculate and output UOd

Input variables, calculate and output PBP and UOBP

Input P

If P ⩽ PBP, calculate and output RSb and UOb

If P > PBP, calculate and output UO

Loop back for new P

Calculate UOd

Calculate UOb

Calculate UO

GAS-OIL RATIO, BUBBLE POINT PRESSURE

```
 01♦LBL "RS"
"GAS/OIL"  26
XROM "TITLE"  FC?C 25
PROMPT  "SCM/M3"
ASTO 03  CLA  ASTO 04
XROM "W8"

 12♦LBL 00
XROM "P"  RCL 14  X<Y?
GTO 01  RDN  XEQ "CRSb"
"RSb"  GTO 02

 21♦LBL 01
RCL 13  "RS"

 24♦LBL 02
XROM "X6"  ADV  GTO 00

 28♦LBL "CRSb"
STO 00  XEQ 03
XEQ "CGS"  *  RCL 00
ST/ Z  R↑  Y↑X  *  ST* Y
RTN

 40♦LBL 03
30  RCL 12  X>Y?  GTO 04
1.0937  ENTER↑  .0362
ENTER↑  25.724  GTO 05

 51♦LBL 04
1.187  ENTER↑  .0178
ENTER↑  23.931

 57♦LBL 05
RCL 12  *  RCL 16  "F-R"
CON  /  E↑X  *  RTN

 67♦LBL "PBP"
"BUBBLE PT"  26
XROM "TITLE"  FC?C 25
PROMPT  "KPA"  ASTO 03
CLA  ASTO 04
XROM "SEPTP"
XROM "OILG"  XROM "W4"
XROM "T"
```

Initialize

Store SI default units for RS in registers 03 and 04

Input variables, calculate and output GAS GS and PBP

Input P

If $P \leq PBP$, calculate and output RSb

If $P > PBP$, output RS

Loop back for new P

Calculate RSb from A, B, C, and OIL G/T′

A, B, and C for OIL G ≤ 30 API

A, B, and C for OIL G > 30 API

Calculate OIL G/T′

Initialize

Store SI default units for PBP in registers 03 and 04

Input variables, calculate and output GAS GS

```
 81↑LBL 06
XROM "IRS"  XEQ "CPBP"
STO 14  "PBP"  XROM "X7"
ADV  GTO 06

 89↑LBL "CPBP"
XEQ 03  XEQ "CGS"  *
RCL 13  X<>Y  /  X<>Y
1/X  Y↑X  END
```

Input RS
Calculate and output PBP
Loop back for new RS

Calculate PBP from A, B, C, and OIL G/T'

TWO-PHASE FORMATION VOLUME FACTOR

```
 01↑LBL "BT"
"2PH VOL FACT"  26
XROM "TITLE"  FC?C 25
PROMPT  XROM "W7"
XEQ "CBOb"  STO 07
"BTBP"  GTO 02

 12↑LBL 00
XROM "P"  RCL 14  X<Y?
GTO 01  RDN  XEQ "CBTb"
STO 06  X<>Y  XROM "X8"
RCL 06  "BTb"  GTO 02

 25↑LBL 01
RCL 07  XEQ "CBT"  "BT"

 29↑LBL 02
FS? 08  ADV  XROM "OUT"
ADV  CF 08  GTO 00

 36↑LBL "CBTb"
STO 05  RCL 11  /
RCL 16  "F-R"  CON
RCL 10  /  X<>Y
XEQ "CBG"  "FT3-BBL"
CON  X<> 05  XEQ "CBOb"
R↑  STO Z  RCL 13  -
RCL 05  *  -  END
```

Initialize

Input variables, calculate and output GAS GS, PBP, and BTBP

Input P
If P $\leq$ PBP, calculate and output RSb and BTb

If P $>$ PBP, calculate and output BT

Loop back for new P

Calculate BTb or BTBP

WATER ISOTHERMAL COMPRESSIBILITY

```
 01♦LBL "CW"
"H2O ISO CMP"  20
XROM "TITLE"  FC?C 25
PROMPT  "1/KPA"
XROM "W6"  XROM "%NACL"
XROM "T"

 11♦LBL 00
XROM "P"  XEQ "CCW"
"CW"  XROM "X2"  ADV
GTO 00

 18♦LBL "CCW"
FC? 06  GTO 01
XEQ "CRSW"  RCL Z  /
89 E-4  *  1  +

 28♦LBL 01
RCL 17  -88 E-11  *
39267 E-9  +  RCL 16  *
RCL 17  477 E-9  *
.01052  -  +  RCL 16  *
RCL 17  134 E-6  *
3.8546  -  -  FS? 06  *
1 E6  /  RCL 19  X≠0?
GTO 02  X<>Y  RTN

 59♦LBL 02
X<>Y  RCL 16  1.121 E-9
*  114 E-8  -  RCL 16  *
27 E-5  +  RCL 16  *
.052  -  R↑  .7  Y↑X  *
1  +  *  END
```

Initialize
Store SI default units for CW in registers 03 and 04
Flag 06: Gas-saturated or gas-free water
Input variables

Input P
Calculate and output CW
Loop back for new P

If gas-saturated, calculate correction factor

Calculate CW

If gas-saturated, multiply by correction factor
If no salt, return

Calculate salinity correction

WATER FORMATION VOLUME FACTOR

```
 01♦LBL "BW"
"H2O VOL FACT"  20
XROM "TITLE"  FC?C 25
PROMPT  XROM "W6"
XROM "%NACL"  XROM "T"

 10♦LBL 00
XROM "P"  XEQ "CBW"
FS? 08  ADV  "BW"
XROM "OUT"  ADV  CF 08
GTO 00

 20♦LBL "CBW"
XEQ 07  RCL 17  *
XEQ 04  +  RCL 17  *
XEQ 02  +  RCL 19  X≠0?
GTO 01  X<>Y  RTN

 35♦LBL 01
X<>Y  RCL 17  85 E-14  *
3.23 E-8  -  RCL 16  60
-  *  RCL 17  -195 E-12
ST* Y  CLX  547 E-8
ST+ Y  RDN  ST+ Y  CLX
LASTX  *  RCL 17  51 E-9
*  +  RCL 19  *  1  +  *
RTN

 67♦LBL 02
FS? 06  GTO 03  .9947
ENTER↑  58 E-7  ENTER↑
102 E-8  GTO 05

 76♦LBL 03
.9911  ENTER↑  635 E-7
ENTER↑  85 E-8  GTO 09

 83♦LBL 04
FS? 06  GTO 06
-4228 E-9  ENTER↑
1.8376 E-8  ENTER↑
-677 E-13

 91♦LBL 05
GTO 09
```

Initialize
Flag 06: Gas-saturated or gas-free water
Input variables

Input P
Calculate and output BW

Loop back for new P

Calculate BW

$BW = A + BP + CP^2$

If no salt, return

Calculate salinity correction

A coefficients for gas-free water

A coefficients for gas-saturated water

B coefficients for gas-free water

WATER FORMATION VOLUME FACTOR (cont.)

```
 93♦LBL 06
-1093 E-9  ENTER↑
-3.497 E-9  ENTER↑
457 E-14  GTO 09

100♦LBL 07
FS? 06  GTO 08  13 E-11
ENTER↑  -13855 E-16
ENTER↑  4285 E-18
GTO 09

109♦LBL 08
-5 E-11  ENTER↑
6429 E-16  ENTER↑
-143 E-17

115♦LBL 09
R↑  STO L  CLX  RCL 16
ST* Y  X<> L  RDN  +
RCL 16  *  +  END
```

B coefficients for gas-saturated water

C coefficients for gas-free water

C coefficients for gas-saturated water

Calculate A, B, and C

WATER VISCOSITY

```
 01♦LBL "UW"
"H2O VIS"  20
XROM "TITLE"  FC?C 25
PROMPT  "PA*S"  ASTO 03
CLA  ASTO 04
XROM "%NACL"  XROM "T"

 13♦LBL 00
XROM "P"  XEQ "CUW"
"UW"  XROM "X3"  ADV
GTO 00

 20♦LBL "CUW"
RCL 16  XEQ "CPSAT"
RCL 17  X<>Y  X<=Y?
GTO 02

 27♦LBL 01
"P < PSAT"  PROMPT
GTO 01
```

Initialize

Store SI default units for UW in registers 03 and 04

Input variables

Input P

Calculate and output UW

Loop back for new P

If $P \geq PSAT$, continue

If $P < PSAT$, halt with error message

```
 31↑LBL 02
-  "PSI-BAR"  CON
1.0467 E-6  *  RCL 16
"F"  CON  305  -  *  1
+  241.4  *  247.8
RCL 16  "F"  CON  140  -
/  4  -  10↑X  *  RCL 19
X≠0?  GTO 04  X<>Y  RTN
```

Calculate UW

If no salt, return

```
 63↑LBL 04
SQRT  344 E-6  *  .00276
-  RCL 19  *  .0135
RCL 16  *  LASTX  SQRT
-  *  RCL 19  X↑2
218 E-6  *  .00187  -
RCL 19  SQRT  *  +  1  +
*  RTN
```

Calculate salinity correction

```
 92↑LBL "CPSAT"
65  X<>Y  "F-C"  CON
STO Z  -  100  /  STO Z
5.218684 E-4  *
.002520658  +  R↑  *
.00439993  +  R↑  *
.001094098  +  R↑  *
-.008685635  +  R↑  *
-.1155286  +  R↑  *
-.29721  +  R↑  *
7.419242  -  374.136
RCL Z  -  LASTX  "C"
CON  /  *  E↑X  22088 E3
*  "PSI"  INCON  END
```

Calculate PSAT

GAS-WATER RATIO

```
 01♦LBL "RSW"
"GAS/WATER"  20
XROM "TITLE"  FC?C 25
PROMPT  "SCM/M3"
ASTO 03  CLA  ASTO 04
XROM "%NACL"  XROM "T"

 13♦LBL 00
XROM "P"  XEQ "CRSW"
"RSW"  XROM "X6"  ADV
GTO 00

 20♦LBL "CRSW"
-102 E-13  RCL 16  *
3.9 E-9  +  RCL 16  *
875 E-9  -  STO 00
ST+ 00  RCL 17  ST* 00
*  148 E-9  RCL 16  *
526 E-7  -  RCL 16  *
.0107  +  ST+ 00  +
RCL 17  *  -359 E-7
RCL 16  *  .00345  +
RCL 16  *  2.12  +  +
RCL 00  X<>Y  173 E-6
RCL 16  *  .0753  -
RCL 19  *  1  +  STO T
ST* Z  *  END
```

Initialize

Store SI default units for RSW in registers 03 and 04

Input variables

Input P

Calculate and output RSW

Loop back for new P

Calculate RSW

Calculate salinity correction

ROCK COMPRESSIBILITY

```
 01♦LBL "CFR"
"ROCK CMP"  19
XROM "TITLE"  FC?C 25
PROMPT  "1/KPA"  ASTO 03
CLA  ASTO 04

 11♦LBL 00
XROM "%POR"  XEQ "CCFR"
"CFR"  XROM "X2"  ADV
GTO 00

 18♦LBL "CCFR"
RCL 18  100  /  -.415
Y↑X  187 E-8  *  END
```

Initialize

Store SI default units for CFR in registers 03 and 04

Input %POR

Calculate and output CFR

Loop back for new %POR

Calculate CFR

TOTAL ISOTHERMAL COMPRESSIBILITY

```
 01♦LBL "CT"
"TOT ISO CMP"  26
XROM "TITLE"  FC?C 25
PROMPT  "1/KPA"
XROM "W6"  XROM "ITcPc"
XROM "STDTP"
XROM "SEPTP"
XROM "OILG"  XROM "GASG"
XROM "%NACL"
XROM "%POR"  XROM "T"
XROM "RSI"  XEQ "CPBP"
STO 14

 20♦LBL 00
19  STO 00  "%SO"
XROM "IN"  "%SW"
XROM "IN"  RCL 20  +
100  X<>Y  -  "%SG"
XROM "OUT"  RCL 20  X≠0?
FC? 08  GTO 01
XROM "W5"  XROM "W9"

 40♦LBL 01
XROM "P"  RCL 20  X=0?
GTO 02  RDN  RCL 14
X<Y?  GTO 02  RDN
XEQ "CCTb"  STO 06  X<>Y
XROM "X8"  RCL 06  "CTb"
GTO 03

 57♦LBL 02
RDN  XEQ "CCT"  "CT"

 61♦LBL 03
XROM "X2"  ADV  GTO 00

 65♦LBL "CCT"
CLX  STO 06  RCL 20
X≠0?  GTO 04  RDN  RDN
GTO 05

 74♦LBL 04
XEQ "CCO"  RCL 20  *
STO 06  GTO 06
```

Initialize

Store SI default units for CT in registers 03 and 04
Flag 06: Gas-saturated or gas-free water

Input variables

Calculate PBP

Input %SO and %SW, calculate %SG

If oil present and 1st pass, calculate and output GAS GS and PBP

Input P
If oil present and P > PBP, calculate and output CT
If oil present and P ≤ PBP, calculate and output RSb and CTb

If no oil present, calculate and output CT

Loop back for new %SO, %SW, and P

Calculate CT

If oil present, calculate CO
Ignore free gas if oil present

TOTAL ISOTHERMAL COMPRESSIBILITY (cont.)

```
 80♦LBL "CCTb"
RDN  STO 07  X<>Y
STO 06  X<>Y  R↑
XEQ "CCOb"  X<>Y  STO 05
X<>Y  RCL 20  *  X<> 06
RCL 07

 95♦LBL 05
100  RCL 20  -  RCL 21
-  X=0?  GTO 06  STO 07
RDN  XEQ "CCG"  RCL 07
*  ST+ 06

109♦LBL 06
RCL 21  X=0?  GTO 07
XEQ "CCW"  RCL 21  *
ST+ 06

117♦LBL 07
RCL 05  XEQ "CCFR"
RCL 06  100  /  +  END
```

Calculate CTb

Since oil present, calculate COb

If gas present, calculate CG
(This free gas ignored if oil present and P > PBP)

If water present, calculate CW

Calculate CFR

GENERAL PURPOSE AND INPUT SUBROUTINES

```
 01♦LBL "TITLE"
SF 08  SF 12  SF 21  ADV
FS? 55  PRA  CF 12  ADV
"SIZE>="  ARCL X  1  -
SF 25  RCL IND X  RTN

 17♦LBL "W6"
ASTO 03  CLA  ASTO 04
"RSW>0"  6

 23♦LBL "Y/N?"
CF 23  STO 00  ASTO 05
"├? Y/N:"  FS? IND X
"├Y"  FC? IND X  "├N"
AON  PROMPT  AOFF
FC? 23  RTN  CF IND 00
ASTO X  "Y"  ASTO Y
X=Y?  SF IND 00  FC? 55
RTN  CLA  ARCL 05  "├: "
FS? IND 00  "├YES"
FC? IND 00  "├NO"  PRA
RTN
```

Initialize

Check size

Flag 06: Gas-saturated or gas-free water

Initialize

Build question and display it
If no ALPHA input, return

If ALPHA input, set flag accordingly

Build response and print it

```
 54♦LBL "ITcPc"
9  XEQ 01  XROM "INU"
"Pc"

 59♦LBL 00
XEQ 05  XROM "INU"  RTN

 63♦LBL 01
"Tc"

 65♦LBL 02
STO 00  ASTO T  "R"
ASTO 01  "K"  ASTO Y
GTO 06

 73♦LBL "STDTP"
RCL 22  SF 25  X=0?  60
STO 22  RCL 23  SF 25
X=0?  14.65  STO 23  21
"STD T"  XEQ 03  "STD P"
GTO 00

 89♦LBL "SEPTP"
23  "SEP T"  XEQ 03
"SEP P"  GTO 00

 95♦LBL "T"
15  "T"

 98♦LBL 03
XEQ 04  XROM "INU"  RTN

102♦LBL 04
STO 00  ASTO T  "F"
ASTO 01  "C"  ASTO Y
GTO 06

110♦LBL 05
ASTO T  "PSI"  ASTO 01
"KPA"  ASTO Y

116♦LBL 06
CLA  ASTO 02  ASTO Z
ARCL T  RTN
```

Input Tc

Input Pc

Tc name

1st six chars. of units for Tc

Store default values for STD T and STD P

Input STD T and STD P

Input SEP T and SEP P

Input T

Get units and input T

1st six chars. of units for T, STD T, and SEP T

1st six chars. of units for Pc, P, STD P, SEP P

Last six chars. of units for temperature and pressure

GENERAL PURPOSE AND INPUT SUBROUTINES (cont.)

```
122♦LBL "P"
16  STO 00  "P"  XEQ 00
RCL 16  "F-R"  SF 25
CON  RCL 10  SF 25  /
RCL 17  RCL 11  SF 25  /
CF 25  RCL 17  RTN

141♦LBL "GASG"
14  STO 00  "GAS G"
XROM "IN"  FS? 22  RTN
28.964  "MW"  XEQ 09
STO 15  RTN

153♦LBL "OILG"
11  STO 00  "API"
ASTO 01  CLA  ASTO 02
ASTO Z  "KG/M3"  ASTO Y
"OIL G"  XROM "INU"  RTN

166♦LBL "IRS"
"RS"  GTO 07

169♦LBL "RSI"
"RSI"

171♦LBL 07
12  XEQ 08  XROM "INU"
RTN

176♦LBL 08
STO 00  ASTO T
"SCF/BBL"  ASTO 01  ASHF
ASTO 02  "SCM/M3"
ASTO Y  CLA  ASTO Z
ARCL T  RTN

189♦LBL "%NACL"
18  STO 00  "%NACL"
XROM "IN"  FS? 22  RTN
1 E4  "PPM"  XEQ 09
STO 19  RTN
```

Input P

Try to calculate TR and PR

Input GAS G
If GAS G input, return
Input MW

Input OIL G

Input RS

Input RSI

Get units and input RS or RSI

Units for RS, RSI, RSb

Input %NACL

If %NACL input, return
Input PPM

```
201♦LBL 09
STO 02  SF 25  *  CF 25
STO 01  0  STO 00
XROM "IN"  RCL 02  /
RTN

213♦LBL "%POR"
17  STO 00  "%POR"
XROM "IN"  RTN

219♦LBL "W7"
XROM "ITcPc"
XROM "STDTP"

222♦LBL "W8"
XROM "SEPTP"
XROM "OILG"  XROM "W4"
XROM "T"  XROM "RSI"

228♦LBL "W9"
XEQ "CPBP"  STO 14
"PBP"

232♦LBL 10
XEQ 05  XROM "OUTU"  RTN

236♦LBL "X0"
STO 11  RDN  STO 10  RDN
STO 06  LASTX  STO 07
RCL Z  XEQ 01
XROM "OUTU"  RCL 06
"Pc"  XEQ 10  RCL 07
X=0?  RTN  "CWA"  XEQ 04
XROM "OUTU"  RCL 10
"Tc*"  XEQ 02
XROM "OUTU"  RCL 11
"Pc*"  GTO 10

263♦LBL "X1"
ASTO T  "FT3/SCF"
GTO 11

267♦LBL "X2"
ASTO T  "1/PSI"  GTO 11
```

Try to convert A to B
Input B
Convert B to A
A = GAS G or %NACL, B = MW or PPM

Input %POR

Input Tc, Pc, STD T, and STD P

Input SEP T, SEP P, OIL G, and GAS G
Calculate and output GAS GS

Calculate PBP

Get units and output PBP

Store Pc*, Tc*, Pc, and CWA

Output Tc and Pc

If CWA = 0, return
If CWA ≠ 0, output CWA, Tc*, and Pc*

English units for BG

English units for CG, CO, CW, CFR, and CT

GENERAL PURPOSE AND INPUT SUBROUTINES (cont.)

```
271♦LBL "X3"
ASTO T  "CP"  GTO 11

275♦LBL "X4"
ASTO T  "BTU/SCF"
GTO 12

279♦LBL "X5"
ASTO T  "BTU/LBM*F"
GTO 12

283♦LBL "X6"
ASTO T  "SCF/BBL"
GTO 11

287♦LBL "X7"
ASTO T  "PSI"

290♦LBL 11
FS? 08  ADV

293♦LBL 12
ASTO 01  ASHF  ASTO 02
CLA  ARCL T  RCL 04
RCL 03  RCL Z
XROM "OUTK"  RDN  STO 03
RDN  STO 04  CF 08  RTN

309♦LBL "X8"
"RSb"  XEQ 08
XROM "OUTU"  END
```

English units for UG, UO, and UW

English units for NHV, GHVD, and GHVW

English units for CP and CV

English units for RS, RSb, and RSW

English units for PBP

Space for readability

Get user's units

Output variable
Save user's units

Get units and output RSb

I/O SUBROUTINES

```
 01♦LBL "OUTK"
SF 22  GTO 00

 04♦LBL "OUTU"
CF 22

 06♦LBL 00
AOFF  STO 00  RDN
ASTO 05  FS? 08  GTO 00
FS? 22  GTO 01

 15♦LBL 00
FS? 09  GTO 01  RCL 02
RCL 01

 20♦LBL 01
CF 22  FS? 10  GTO 02
CLA  ARCL X  ARCL Y
GTO 04

 28♦LBL 02
RCL 00  XEQ 13  INCON
CF 25

 33♦LBL 03
CLA  ARCL 05  "├, "
ARCL Y  ARCL Z  "├?"
XEQ 14  AON  TONE 7
STOP  AOFF

 45♦LBL 04
RCL 00  XEQ 12  INCON
FC?C 25  GTO 03  CLA

 52♦LBL 05
ARCL 05  "├="  X<> 00
ARCL 00  FS? 55  GTO 15
X<> 00  RDN  STO 01
X<>Y  STO 02  X<>Y  R↑
XEQ 14  STOP  AOFF
RCL 02  RCL 01  RCL 00
GTO 09
```

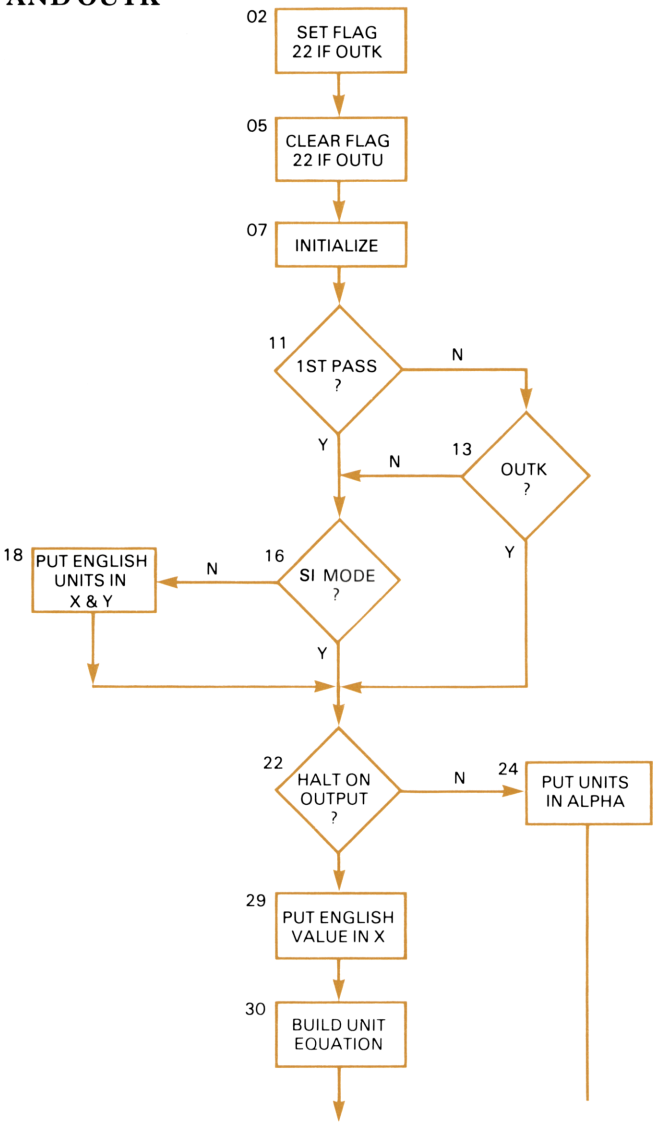
"OUT", "OUTU", "OUTK" I/O subroutines—see flowcharts

```
 73♦LBL "OUT"
AOFF  STO 00  ASTO 05
CLA  ASTO Y  ASTO Z
GTO 05

 81♦LBL "INK"
SF 22  GTO 00

 84♦LBL "INU"
CF 22

 86♦LBL 00
AOFF  RDN  ASTO 05
CF 23  ISG 00  CLD
FS? 08  GTO 06  FS? 22
GTO 07

 97♦LBL 06
FS? 09  GTO 07  RCL 02
RCL 01  RCL IND 00
GTO 08

104♦LBL 07
RCL IND 00  XEQ 13
INCON  CLA  ARCL 05

110♦LBL 08
CF 22  CF 25  "⊢=?"
XEQ 14  STOP  AOFF
FC? 22  FS? 23  GTO 11
RCL IND 00  ASTO Y  ASHF
ASTO Z

124♦LBL 09
CF 25  CF 21  FS? 55
SF 21  RTN

130♦LBL 10
CLA  ARCL Y  ARCL Z
"⊢?"  XEQ 14  AON  STOP
AOFF
```

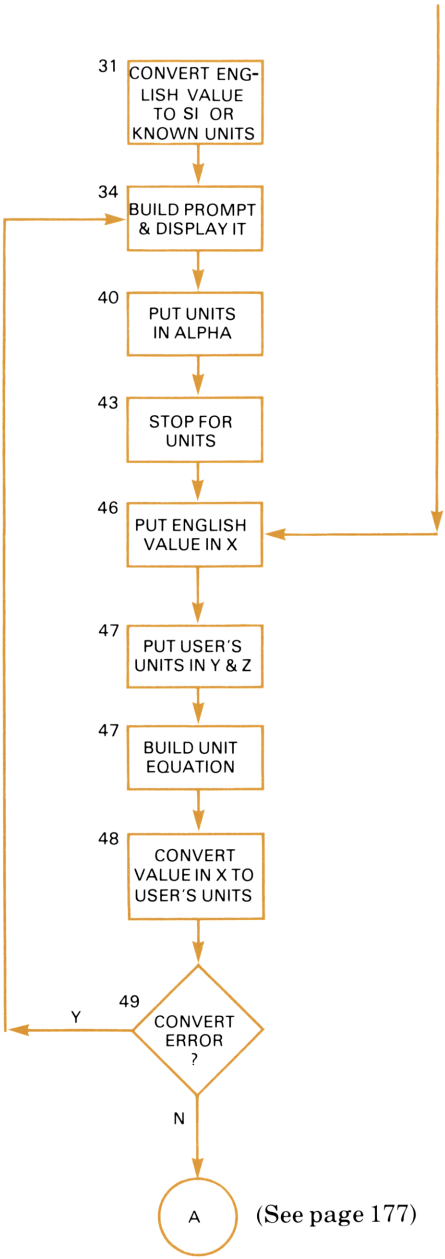
"INU", "INK" I/O subroutines—see flowcharts

```
139♦LBL 11
XEQ 12  CON  FC?C 25
GTO 10  STO IND 00
CF 21  FC? 55  RTN  CLA
ARCL 05  "⊢="  ARCL L

152♦LBL 15
"⊢ "  ARCL Y  ARCL Z
SF 21  PRA  RTN

159♦LBL 12
ASTO Y  ASHF  ASTO Z

163♦LBL 13
SF 25  CLA  ARCL Y
ARCL Z  "⊢-"  ARCL 01
ARCL 02  RTN

172♦LBL 14
CF 21  AVIEW  CLA
ARCL Y  ARCL Z  FS? 55
SF 21  END
```

OUTU AND OUTK

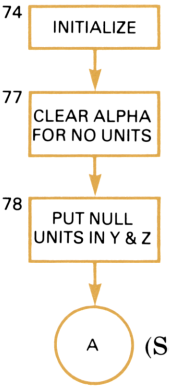


Note: Small numbers are the line numbers in the program listing.



Note: Small numbers are the line numbers in the program listing.

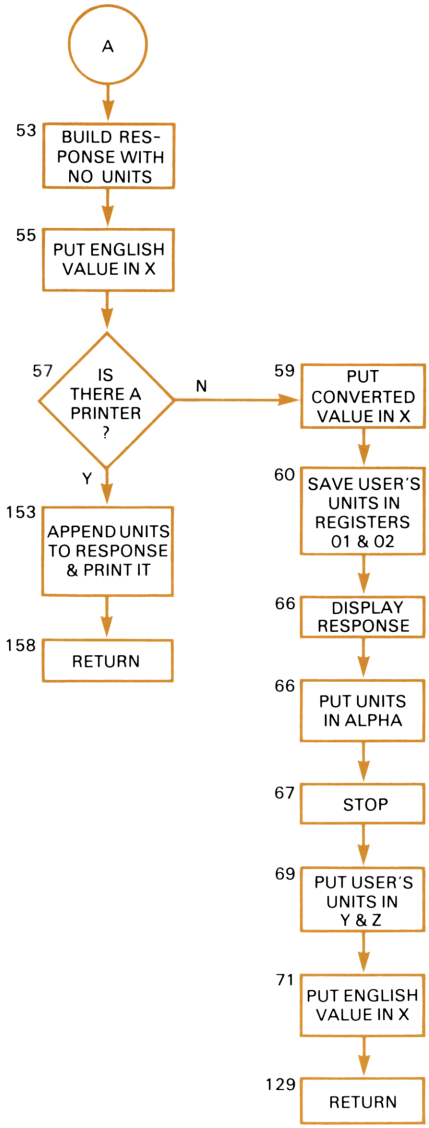
OUT



(See page 177)

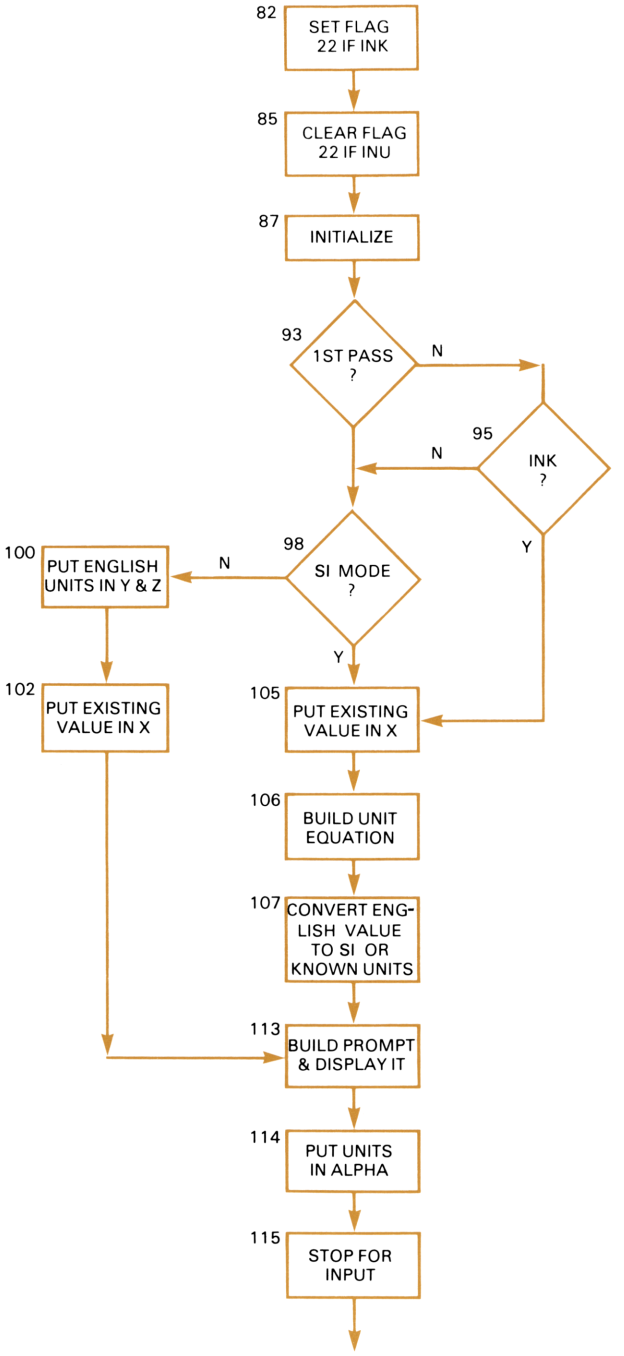
Note: Small numbers are the line numbers in the program listing.

OUT, OUTU, AND OUTK (cont.)

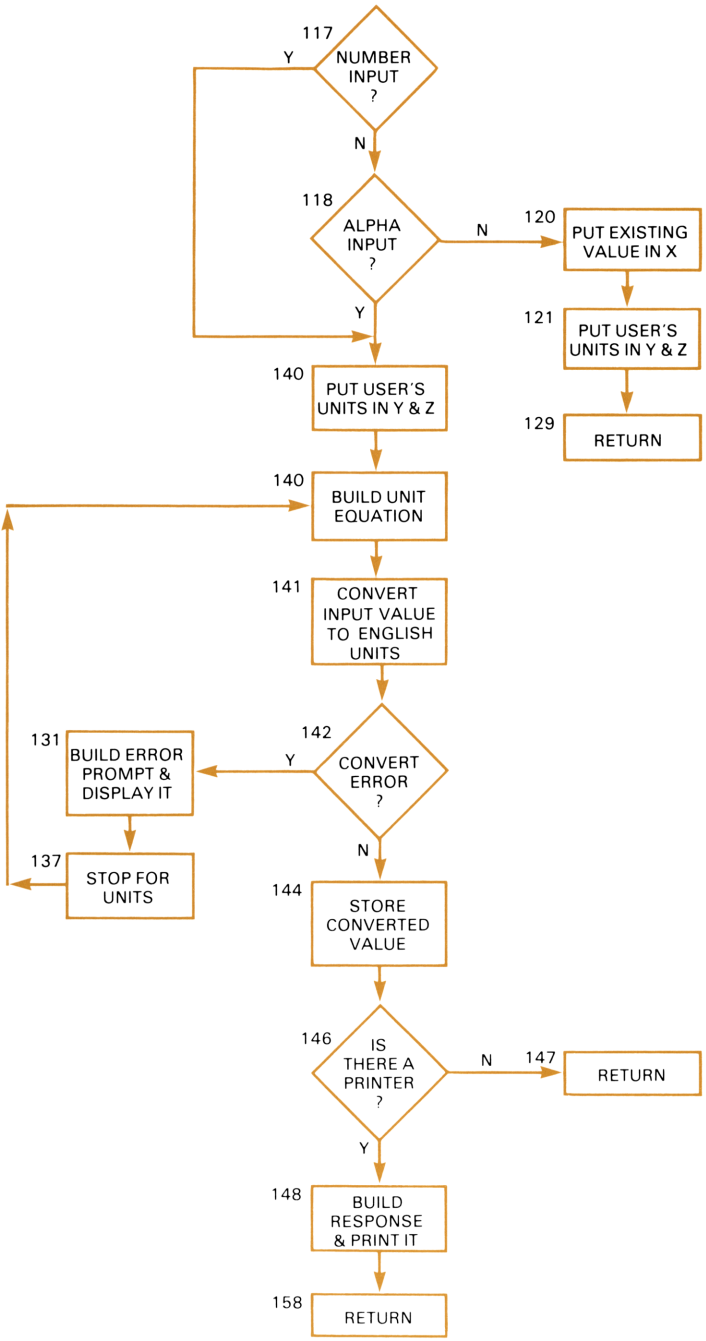


Note: Small numbers are the line numbers in the program listing.

INU AND INK



Note: Small numbers are the line numbers in the program listing.



Note: Small numbers are the line numbers in the program listing.

APPENDIX A

PROGRAM AND CALCULATION SUBROUTINE DATA

Legend:

1. Program or Calculation Subroutine Name
2. Number of Registers to Copy Program
3. Minimum Size Required to Run Program
4. Variables Required in Proper Registers (see Appendix B)
5. Stack Contents When Subroutine Called*
6. Stack Contents When Subroutine Returns*
7. Which Output Variables Are Stored by Program
8. Which Subroutines Are Called
9. Number of Subroutine Levels Used by Subroutine†
10. Scratch Registers Used by Subroutine‡
11. Flags Used**

*All inputs to a calculation subroutine, whether in registers or in the stack, must be in **Pac English default units**. All values returned in the stack by a calculation subroutine will be in **Pac English default units**.

†The number of subroutine levels used includes one level used by the calling program to call the subroutine. This means that a subroutine needing 6 subroutine levels could be called **as** a subroutine, but not **from** a subroutine.

‡Registers 00-05 are used by the calling program for input, output, and units. See Appendix B.

** "Flags Used" does not include the flags used for input and output, printer formatting, and error checking: 08-10, 12, 21-23, 25, 55.

1	2	3	4	5	6	7	8	9	10	11
Z Factor (Z)	7	018	Tc, Pc, T, P	—	—	—	TITLE, ITcPc, T, P, CZ, OUT	—	—	—
Calculate Z (CZ)††	—	—	—	Y=TR X=PR	Y=TR X=Z L=PR	—	—	0	—	—
Gas Isothermal Compressibility (CG)	11	018	Tc, Pc, T, P	—	—	—	TITLE, ITcPc, T, P, CCG, OUTK	—	—	—
Calculate CG (CCG)	—	—	Pc	Y=TR X=PR	X=CG	—	CCR	1	—	—
Calculate Pseudoreduced Compressibility (CCR)††	—	—	—	Y=TR X=PR	Y=TR X=CR L=PR	—	—	0	—	—
Gas Formation Volume Factor (BG)	14	024	Tc, Pc, STD T, STD P, T, P	—	—	—	TITLE, ITcPc, STDTP, T, P, CBG, OUTK	—	—	—
Calculate BG (CBG)	—	—	Pc, STD T, STD P, T	Y=TR X=PR	X=BG	—	CZ	1	—	—

†† Because these are machine language functions, you will not be able to copy, list, or single step through them.

Legend:

1. Program or Calculation Subroutine Name
2. Number of Registers to Copy Program
3. Minimum Size Required to Run Program
4. Variables Required in Proper Registers (see Appendix B)
5. Stack Contents When Subroutine Called
6. Stack Contents When Subroutine Returns
7. Which Output Variables Are Stored by Program
8. Which Subroutines Are Called
9. Number of Subroutine Levels Used by Subroutine
10. Scratch Registers Used by Subroutine
11. Flags Used

1	2	3	4	5	6	7	8	9	10	11
Gas Viscosity (UG)	22	018	Tc, Pc, GAS G, T, P	—	—	—	TITLE, ITcPc, GASG, T, P, CUG, OUTK	—	00, 05	—
Calculate UG (CUG)	—	—	GAS G, T, P	Y=TR X=PR	X=UG	—	CZ	1	00, 05	—
Pseudocritical Temperature and Pressure From Gas Gravity (TcPc)	51	029	GAS G, %N2, %CO2, %H2S	—	—	Tc*, Pc*	TITLE, Y/N?, GASG, SOUR, CTcPc, OUTU	—	00, 01, 05	05
Calculate Tc and Pc (CTcPc)	—	—	GAS G, %N2, %CO2, %H2S	—	T=Tc Z=Pc Y=Tc* X=Pc* L=CWA	—	CCWA	2	00, 01, 05	05

1	2	3	4	5	6	7	8	9	10	11
Calculate Wichert-Aziz Correction (CCWA)	—	—	%CO2, %H2S	Y=Tc X=Pc	T=Tc Z=Pc Y=Tc* X=Pc* L=CWA	—	—	1	00, 01, 05	—
Gas Properties From Composition (PROP)	217	045	%N2-%H2O, GAS G, T	—	—	GAS G, Tc*, Pc*	TITLE, Y/N?, COMP, OUT, CGASG, CTPC, OUTU, CHV, OUTK, T, CCK	—	00, 01, 05, 06, 07	04, 07, 27
Calculate GAS G (CGASG)	—	—	%N2-%H2O	—	X=GAS G	—	—	2	00	—
Calculate Tc and Pc From Composition (CTPC)	—	—	%N2-%H2O	—	T=Tc Z=Pc Y=Tc* X=Pc* L=CWA	—	CCWA	2	00, 01, 05	—
Calculate Heating Values (CHV)	—	—	%N2-%H2O	—	Z=NHV Y=GHVD X=GHVW	—	—	2	00	—
Calculate CP, CV, and K (CCK)	—	—	%N2-%H2O, GAS G, T	—	Z=CP Y=CV X=K	—	—	2	00	—

Legend:

1. Program or Calculation Subroutine Name
2. Number of Registers to Copy Program
3. Minimum Size Required to Run Program
4. Variables Required in Proper Registers (see Appendix B)
5. Stack Contents When Subroutine Called
6. Stack Contents When Subroutine Returns
7. Which Output Variables Are Stored by Program
8. Which Subroutines Are Called
9. Number of Subroutine Levels Used by Subroutine
10. Scratch Registers Used by Subroutine
11. Flags Used

1	2	3	4	5	6	7	8	9	10	11
Oil Isothermal Compressibility (CO)	35	026	Tc, Pc, STD T, STD P, SEP T, SEP P, OIL G, GAS G, T, RSI, P	—	—	PBP	TITLE, ITcPc, STDTP, SEPTP, OILG, GASG, CGS, OUT, T, RSI, CPBP, OUTU, P, CRSb, CCOb, CCO, OUTK	—	00, 01, 02, 05	—
Calculate CO Below Bubble Point (CCOb)	—	—	STD T, STD P, SEP T, SEP P, OIL G, GAS G, T	Z=TR Y=PR X=P	Y=RSb X=COb	—	CBG, CBOb	4	00, 01, 02, 05	—

1	2	3	4	5	6	7	8	9	10	11
Calculate CO Above Bubble Point (CCO)	—	—	SEP T, SEP P, OIL G, GAS G, T, RSI, P	—	X=CO	—	CGS	2	—	—
Calculate GAS GS (CGS)	—	—	SEP T, SEP P, OIL G, GAS G	—	X=GAS GS	—	—	1	—	—
Oil Formation Volume Factor (BO)	33	026	SEP T, SEP P, OIL G, GAS G, T, RSI, P, PBP	—	—	PBP	TITLE, SEPTP, OILG, GASG, CGS, OUT, T, RSI, CPBP, OUTU, P, CRSb, CBOb, CBO	—	00, 01, 02, 06, 07	—
Calculate BO Below Bubble Point (CBOb)	—	—	SEP T, SEP P, OIL G, GAS G, T	X=P	T=RSb Z=∂RSb/∂P Y=∂BOb/∂RSb X=BOb	—	CRSb, CGS	3	00, 01, 02	—
Calculate BO Above Bubble Point (CBO)	—	—	SEP T, SEP P, OIL G, GAS G, T, RSI, PBP, P	X=PBP X=BOBP	X=BOBP X=BO	—	CCO	3	—	—

Legend:

1. Program or Calculation Subroutine Name
2. Number of Registers to Copy Program
3. Minimum Size Required to Run Program
4. Variables Required in Proper Registers (see Appendix B)
5. Stack Contents When Subroutine Called
6. Stack Contents When Subroutine Returns
7. Which Output Variables Are Stored by Program
8. Which Subroutines Are Called
9. Number of Subroutine Levels Used by Subroutine
10. Scratch Registers Used by Subroutine
11. Flags Used

1	2	3	4	5	6	7	8	9	10	11
Oil Viscosity (UO)	36	026	SEP T, SEP P, OIL G, GAS G, T, RSI, P	—	—	—	TITLE, SEPTP, OILG, GASG, CGS, OUT, T, CUOd, OUTK, RSI, CPBP, OUTU, CUOb, P, CRSb, CUO	—	00, 06, 07	—
Calculate UO for Dead Oil (CUOd)	—	—	OIL G, T	—	X=UOd	—	—	1	—	—
Calculate UO for Live Oil Below Bubble Point (CUOb)	—	—	—	Y=RSb X=UOd Y=RSI X=UOd	X=UOb X=UOBP	—	—	1	—	—
Calculate UO for Live Oil Above Bubble Point (CUO)	—	—	PBP, P	X=UOBP	X=UO	—	—	1	—	—

1	2	3	4	5	6	7	8	9	10	11
Gas-Oil Ratio (RS)	35	026	SEP T, SEP P, OIL G, GAS G, T, RSI, P	—	—	PBP	TITLE, SEPTP, OILG, GASG, CGS, OUT, T, RSI, CPBP, OUTU, P, CRSb, OUTK	—	00	—
Calculate RS Below Bubble Point (CRSb)	—	—	SEP T, SEP P, OIL G, GAS G, T	X=P	Y=∂RSb/∂P X=RSb	—	CGS	2	00	—
Bubble Point Pressure (PBP)	35	026	SEP T, SEP P, OIL G, GAS G, T, RS	—	—	PBP	TITLE, SEPTP, OILG, GASG, CGS, OUT, T, IRS, CPBP, OUTK	—	—	—
Calculate PBP (CPBP)	—	—	SEP T, SEP P, OIL G, GAS G. T, RS	—	X=PBP	—	CGS	2	—	—

Legend:

1. Program or Calculation Subroutine Name
2. Number of Registers to Copy Program
3. Minimum Size Required to Run Program
4. Variables Required in Proper Registers (see Appendix B)
5. Stack Contents When Subroutine Called
6. Stack Contents When Subroutine Returns
7. Which Output Variables Are Stored by Program
8. Which Subroutines Are Called
9. Number of Subroutine Levels Used by Subroutine
10. Scratch Registers Used by Subroutine
11. Flags Used

1	2	3	4	5	6	7	8	9	10	11
Two-Phase Formation Volume Factor (BT)	21	026	Tc, Pc, STD T, STD P, SEP T, SEP P, OIL G, GAS G, T, RSI, P	—	—	PBP	TITLE, ITcPc, STDTP, SEPTP, OILG, GASG, CGS, OUT, T, RSI, CPBP, OUTU, P, CRSb, CBTb, CBT	—	00, 01, 02, 05, 06, 07	—
Calculate BT Below Bubble Point (CBTb)	—	—	Tc, Pc, STD T, STD P, OIL G, GAS G, T, RSI, P	X=P X=PBP	Y=RSb X=BTb X=BTBP	—	CBG, CBOb	4	00, 01, 02, 05	—

1	2	3	4	5	6	7	8	9	10	11
Calculate BT Above Bubble Point (CBT)	—	—	SEP T, SEP P, OIL G, GAS G, T, RSI, PBP, P	X=BTBP	X=BT	—	CCO	3	—	—
Water Isothermal Compressibility (CW)	29	020	%NACL, T, P	—	—	—	TITLE, Y/N?, %NACL, T, P, CCW, OUTK	—	00	06
Calculate CW (CCW)	—	—	%NACL, T, P	—	X=CW	—	CRSW	2	00	06
Water Formation Volume Factor (BW)	49	020	%NACL, T, P	—	—	—	TITLE, Y/N?, %NACL, T, P, CBW, OUT	—	—	06
Calculate BW (CBW)	—	—	%NACL, T, P	—	X=BW	—	—	2	—	06
Water Viscosity (UW)	53	020	%NACL, T, P	—	—	—	TITLE, %NACL, T, P, CUW, OUTK	—	—	—
Calculate UW (CUW)	—	—	%NACL, T, P	—	X=UW	—	CPSAT	2	—	—
Calculate PSAT (CPSAT)	—	—	—	X=T	X=PSAT	—	—	1	—	—

Legend:

1. Program or Calculation Subroutine Name
2. Number of Registers to Copy Program
3. Minimum Size Required to Run Program
4. Variables Required in Proper Registers (see Appendix B)
5. Stack Contents When Subroutine Called
6. Stack Contents When Subroutine Returns
7. Which Output Variables Are Stored by Program
8. Which Subroutines Are Called
9. Number of Subroutine Levels Used by Subroutine
10. Scratch Registers Used by Subroutine
11. Flags Used

1	2	3	4	5	6	7	8	9	10	11
Gas-Water Ratio (RSW)	27	020	%NACL, T, P	—	—	—	TITLE, %NACL, T, P, CRSW, OUTK	—	00	—
Calculate RSW (CRSW)	—	—	%NACL, T, P	—	Z=S.C. Y=∂RSW/∂P X=RSW	—	—	1	00	—
Rock Compressibility (CFR)	12	019	%POR	—	—	—	TITLE, %POR, CCFR, OUTK	—	—	—
Calculate CFR (CCFR)	—	—	%POR	—	X=CFR	—	—	1	—	—

1	2	3	4	5	6	7	8	9	10	11
Total Isothermal Compressibility (CT)	38	026	Tc, Pc, STD T, STD P, SEP T, SEP P, OIL G, GAS G, %NACL, %POR, T, RSI, %SO, %SW, P	—	—	PBP	TITLE, Y/N?, ITcPc, STDTP, SEPTP, OILG, GASG, %NACL, %POR, T, RSI, IN, OUT, CGS, CPBP, OUTU, P, CRSb, CCTb, CCT, OUTK	—	00, 01, 02, 05, 06, 07	06
Calculate CT Below Bubble Point (CCTb)	—	—	Pc, STD T, STD P, SEP T, SEP P, OIL G, GAS G, %NACL, %POR, T, %SO, %SW, P	Z=TR Y=PR X=P	Y=RSb X=CTb	—	CCOb, CCG, CCW, CCFR	5	00, 01, 02, 05, 06, 07	06

Legend:

1. Program or Calculation Subroutine Name
2. Number of Registers to Copy Program
3. Minimum Size Required to Run Program
4. Variables Required in Proper Registers (see Appendix B)
5. Stack Contents When Subroutine Called
6. Stack Contents When Subroutine Returns
7. Which Output Variables Are Stored by Program
8. Which Subroutines Are Called
9. Number of Subroutine Levels Used by Subroutine
10. Scratch Registers Used by Subroutine
11. Flags Used

1	2	3	4	5	6	7	8	9	10	11
Calculate CT Above Bubble Point (CCT)	—	—	Pc, SEP T, SEP P, OIL G, GAS G, %NACL, %POR, T, RSI, %SO, %SW, P	Z=TR Y=PR X=P	X=CT	—	CCO, CCW, CCFR	3	00, 06, 07	06

Notes

APPENDIX B

REGISTER CONTENTS AND FLAG USAGE

Register	Contents (units)	Register	Contents (units)
00	Input, Output, Scratch	22	STD T (F)
01	English Units, Scratch	23	STD P (PSI)
02	English Units, Scratch	24	SEP T (F)
03	Known Output Units	25	SEP P (PSI)
04	Known Output Units	26	%N2
05	Input, Output, Scratch	27	%CO2
06	Scratch	28	%H2S
07	Scratch	29	%METH
08	Unused	30	%ETH
09	Unused	31	%PROP
10	Tc, Tc* (R)	32	%IBUT
11	Pc, Pc* (PSI)	33	%N-BUT
12	OIL G (API)	34	%IPEN
13	RS, RSI, RSb (SCF/BBL)	35	%N-PEN
14	PBP (PSI)	36	%N-HEX
15	GAS G	37	%N-HEP
16	T (F)	38	%N-OCT
17	P (PSI)	39	%N-NON
18	%POR	40	%N-DEC
19	%NACL	41	%O2
20	%SO	42	%H2
21	%SW	43	%He
		44	%H2O

Flag	Usage
00-03	Unused
04	Used By: PROP Set: Calculate CP, CV, and K Clear: Don't calculate CP, CV, and K
05	Used By: TcPc, CTcPc Set: Condensate well fluid Clear: Miscellaneous reservoir gas
06	Used By: CW, CCW, BW, CBW, CT, CCTb, CCT Set: Gas-saturated water or brine Clear: Gas-free water or brine
07	Used By: PROP Set: Clear constituent registers Clear: Leave constituent registers unchanged
08	Used By: All programs Set: First pass through the program Clear: Any other pass through the program
09	Used By: INU, INK, OUTU, OUTK Set: Use SI default units Clear: Use English default units
10	Used By: OUTU, OUTK Set: Halt and prompt for units on output Clear: Don't halt on output

The Pac also uses flags 12 (double-wide), 21 (printer enable), 22 (numeric input), 23 (ALPHA input), 25 (error ignore), and 55 (printer existence).

APPENDIX C

INPUT AND OUTPUT VARIABLES

Pac Symbol	Variable Name	Pac English Units*	Pac SI Units
BG	Gas Formation Volume Factor	FT3/SCF	M3/SCM
BO	Oil Formation Volume Factor (above PBP)	—	—
BOBP	Oil Formation Volume Factor (at PBP)	—	—
BOb	Oil Formation Volume Factor (below PBP)	—	—
BT	Two-Phase Formation Volume Factor (above PBP)	—	—
BTBP	Two-Phase Formation Volume Factor (at PBP)	—	—
BTb	Two-Phase Formation Volume Factor (below PBP)	—	—
BW	Water Formation Volume Factor	—	—
CFR	Rock Compressibility	1/PSI	1/KPA
CG	Gas Isothermal Compressibility	1/PSI	1/KPA
CO	Oil Isothermal Compressibility (above PBP)	1/PSI	1/KPA
COb	Oil Isothermal Compressibility (below PBP)	1/PSI	1/KPA
CP	Specific Heat (constant pressure)	BTU/LBM*F	KJ/KG*K
CT	Total Isothermal Compressibility (above PBP)	1/PSI	1/KPA
CTb	Total Isothermal Compressibility (below PBP)	1/PSI	1/KPA
CV	Specific Heat (constant volume)	BTU/LBM*F	KJ/KG*K
CW	Water Isothermal Compressibility	1/PSI	1/KPA
CWA	Wichert-Aziz Correction	F	C
GAS G	Gas Gravity (relative to air)	—	—
GAS GS	GAS G Corrected for Separator Conditions	—	—
GHVD	Gross Heating Value (dry)	BTU/SCF	KJ/SCM

Pac Symbol	Variable Name	Pac English Units*	Pac SI Units
GHVW	Gross Heating Value (wet)	BTU/SCF	KJ/SCM
K	Specific Heat Ratio (CP/CV)	—	—
MW	Molecular Weight (GAS G · 28.964)	—	—
NHV	Net Heating Value	BTU/SCF	KJ/SCM
OIL G	Oil Gravity (relative to water)	API	KG/M3
P	Pressure	PSI	KPA
PBP	Bubble Point Pressure	PSI	KPA
PPM	Parts Per Million (%NACL/10000)	—	—
PR	Reduced Pressure (P/Pc)	—	—
PSAT	Saturation Pressure of Water	PSI	KPA
Pc	Critical or Pseudocritical Pressure	PSI	KPA
Pc*	Pc Corrected for Sour Gas Content	PSI	KPA
RS	Gas-Oil Ratio (above PBP)	SCF/BBL	SCM/M3
RSI	Initial Gas-Oil Ratio (above PBP)	SCF/BBL	SCM/M3
RSW	Gas-Water Ratio	SCF/BBL	SCM/M3
RSb	Gas-Oil Ratio (below PBP)	SCF/BBL	SCM/M3
SEP P	Separator Pressure (absolute P)	PSI	KPA
SEP T	Separator Temperature	F	C
STD P	Pressure at Standard Conditions	PSI	KPA
STD T	Temperature at Standard Conditions	F	C
T	Temperature	F	C
TR	Reduced Temperature (T/Tc)	—	—
Tc	Critical or Pseudocritical Temperature	R	K

*Note that whenever a variable is stored in a register (see Appendix B), it will always be stored in **Pac English default units**.

Pac Symbol	Variable Name	Pac English Units*	Pac SI Units
Tc*	Tc Corrected for Sour Gas Content	R	K
UG	Gas Viscosity	CP	PA*S
UO	Live Oil Viscosity (above PBP)	CP	PA*S
UOBP	Live Oil Viscosity (at PBP)	CP	PA*S
UOb	Live Oil Viscosity (below PBP)	CP	PA*S
UOd	Dead Oil Viscosity	CP	PA*S
UW	Water Viscosity	CP	PA*S
Z	Z Factor	—	—
%CO2	Mole Percent Carbon Dioxide	—	—
%ETH	Mole Percent Ethane	—	—
%He	Mole Percent Helium	—	—
%H2	Mole Percent Hydrogen	—	—
%H2O	Mole Percent Water Vapor	—	—
%H2S	Mole Percent Hydrogen Sulfide	—	—
%IBUT	Mole Percent Isobutane	—	—
%IPEN	Mole Percent Isopentane	—	—
%METH	Mole Percent Methane	—	—
%NACL	Weight Percent Sodium Chloride	—	—
%N2	Mole Percent Nitrogen	—	—
%N-BUT	Mole Percent N-Butane	—	—
%N-DEC	Mole Percent N-Decane	—	—
%N-HEP	Mole Percent N-Heptane	—	—
%N-HEX	Mole Percent N-Hexane	—	—
%N-NON	Mole Percent N-Nonane	—	—
%N-OCT	Mole Percent N-Octane	—	—
%N-PEN	Mole Percent N-Pentane	—	—

Pac Symbol	Variable Name	Pac English Units*	Pac SI Units
%O2	Mole Percent Oxygen	—	—
%POR	Percent Porosity	—	—
%PROP	Mole Percent Propane	—	—
%SG	Volume Percent Gas Saturation	—	—
%SO	Volume Percent Oil Saturation	—	—
%SW	Volume Percent Water Saturation	—	—
%TOT	Total of Mole Percentages	—	—

* Note that whenever a variable is stored in a register (see Appendix B), it will always be stored in **Pac English default units**.

Corvallis Division
1000 N.E. Circle Blvd., Corvallis, OR 97330, U.S.A.

00041-90288 Rev. C English

Printed in U.S.A. 3/83