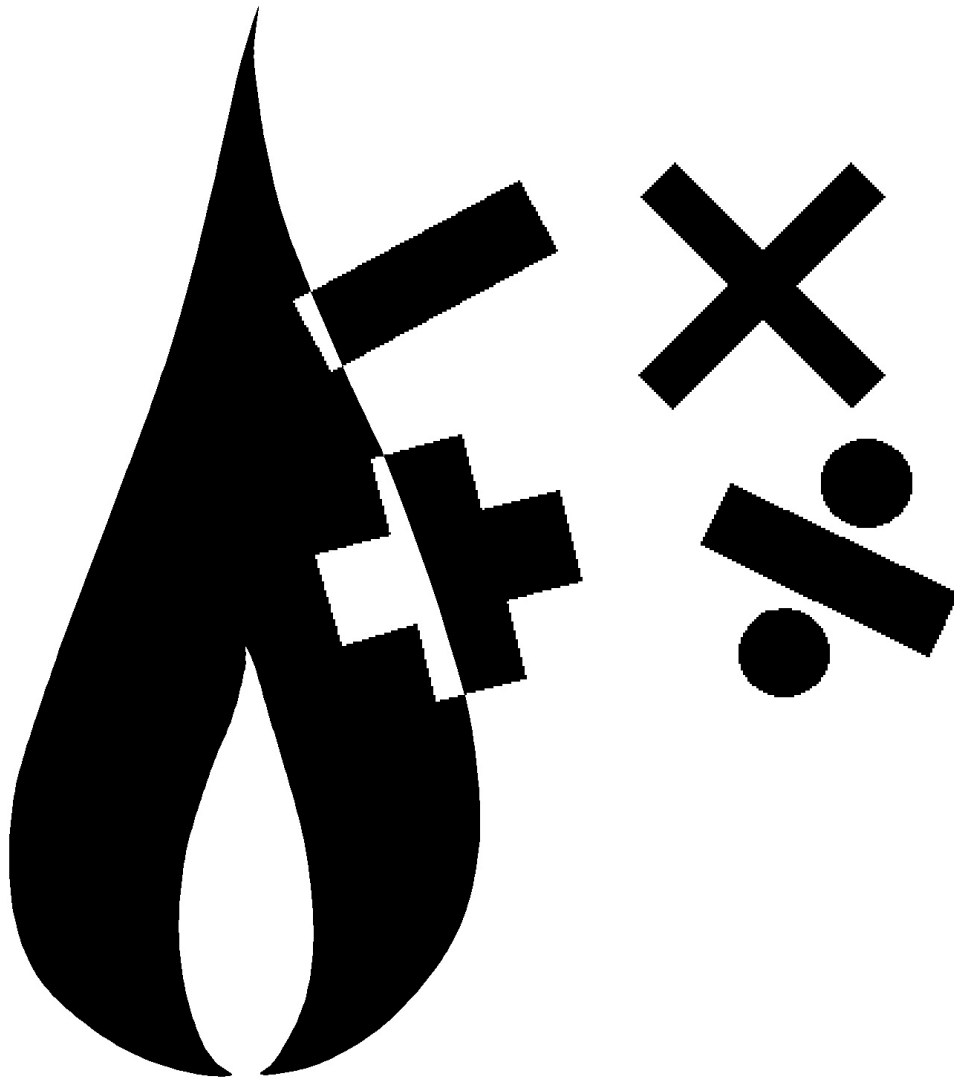


# **GASCalc<sup>TM</sup> API**

## ***Function Reference***



**B3PE LLC**  
**419 East Columbia Street**  
**Colorado Springs, Colorado 80907 USA**  
**(719) 578-9391**  
[www.b3pellc.com](http://www.b3pellc.com)

**© 2025 B3PE LLC - All rights reserved.  
Revision 010**

**GASCalc and the “B-Cubed” logo are trademarks of B3PE LLC.  
Windows is a trademark of Microsoft Corporation.**

|                                                               |                     |
|---------------------------------------------------------------|---------------------|
| General Information .....                                     | <a href="#">1</a>   |
| Modules & Functions. ....                                     | <a href="#">1</a>   |
| API Interface Methods. ....                                   | <a href="#">4</a>   |
| Function Names & Parameters .....                             | <a href="#">5</a>   |
| CalcElev - Atmospheric Elevation Calculation .....            | <a href="#">6</a>   |
| CalcFact - Dimensional Unit Conversion.....                   | <a href="#">8</a>   |
| CalcGP - Gas Properties Calculation .....                     | <a href="#">14</a>  |
| CalcLoss - Gas Loss Calculation .....                         | <a href="#">18</a>  |
| CalcMeter - Displacement Meter Calculation .....              | <a href="#">25</a>  |
| CalcOrifice - Orifice Meter Calculation.....                  | <a href="#">30</a>  |
| CalcPatm - Atmospheric Pressure Calculation.....              | <a href="#">36</a>  |
| CalcPave - Average Pressure Calculation.....                  | <a href="#">38</a>  |
| CalcPipeEquivL - Equivalent Pipe Length Calculation .....     | <a href="#">40</a>  |
| CalcPipeFlow - Pipe Flow Calculation.....                     | <a href="#">42</a>  |
| CalcPipeVelocity - Pipe Velocity Calculation .....            | <a href="#">49</a>  |
| CalcPulse - Pulse Output Meter Calculation .....              | <a href="#">51</a>  |
| CalcPuncture - Equivalent Puncture Diameter Calculation ..... | <a href="#">55</a>  |
| CalcQBase - Base Volume Conversion .....                      | <a href="#">57</a>  |
| CalcQStd - Standard Volume Calculation .....                  | <a href="#">60</a>  |
| CalcReg - Regulator Flow Calculation.....                     | <a href="#">64</a>  |
| CalcRelief - Relief Valve Flow Calculation.....               | <a href="#">69</a>  |
| CalcReynolds - Reynolds Number Calculation .....              | <a href="#">74</a>  |
| CalcSonic - Sonic Velocity Calculation .....                  | <a href="#">77</a>  |
| CalcTherm - Thermodynamic Properties Calculation .....        | <a href="#">81</a>  |
| CalcT - Temperature Unit Conversion .....                     | <a href="#">85</a>  |
| CalcUSM.....                                                  | <a href="#">87</a>  |
| CalcVCone - V-Cone Meter Calculation .....                    | <a href="#">92</a>  |
| CalcZF - Compressibility Calculation .....                    | <a href="#">97</a>  |
| SetLicense - Set License Function .....                       | <a href="#">109</a> |



## General Information

The GASCalc Application Program Interface (API) provides programmatic access to selected calculation routines and methods found in the GASCalc software.

## Modules & Functions

The API is licensed on a module basis. The individual modules allow access to specific calculation methods and routines. The modules are described in the following table:

| Module          | Functions                 | Function Description                            | Methods/Options                                                                                                                                                                                                                                                           |
|-----------------|---------------------------|-------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Compressibility | CalcZF<br>Calc_ZF         | Calculates compressibility factor and density.* | <ul style="list-style-type: none"><li>• AGA 8 - 1985</li><li>• AGA 8 - 1992</li><li>• AGA 8 2017, Detail &amp; Gross</li><li>• AGA 8 2017, GERG - 2008</li><li>• AGA NX19 Standard*</li><li>• GPA 2172-09*</li><li>• ISO 6976 2016</li></ul> * Does not calculate density |
| Gas Properties  | CalcGP<br>Calc_GP         | Calculates various gas property values.         | Heating Value <ul style="list-style-type: none"><li>• AGA 8</li><li>• GPA 2172-86</li><li>• GPA 2172-09</li><li>• GPSA</li><li>• ISO 6976 2016</li></ul>                                                                                                                  |
|                 |                           |                                                 | Ideal Gas Specific Gravity                                                                                                                                                                                                                                                |
|                 |                           |                                                 | Real Gas Specific Gravity                                                                                                                                                                                                                                                 |
|                 |                           |                                                 | Ratio of Specific Heats <ul style="list-style-type: none"><li>• Gas Engineers Handbook</li><li>• GPA</li></ul>                                                                                                                                                            |
|                 |                           |                                                 | Dynamic/Absolute Viscosity                                                                                                                                                                                                                                                |
|                 |                           |                                                 | Speed of Sound (non-AGA 7) <ul style="list-style-type: none"><li>• Traditional Thermodynamic</li></ul>                                                                                                                                                                    |
| General         | SetLicense<br>Set_License | Sets and returns license information.           |                                                                                                                                                                                                                                                                           |

| Module | Functions                     | Function Description                                                                               | Methods/Options                                                                                                                                                                                              |
|--------|-------------------------------|----------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Loss   | CalcLoss<br>Calc_Loss         | Calculates the volumetric flow rate from a punctured or severed pipe.                              |                                                                                                                                                                                                              |
|        | CalcPuncture<br>Calc_Puncture | Calculates the equivalent diameter of an irregular or rectangular shape.                           |                                                                                                                                                                                                              |
| Meter  | CalcOrifice<br>Calc_Orifice   | Calculates various values associated with flow through an orifice meter.                           | <ul style="list-style-type: none"> <li>• AGA 3 - 1985</li> <li>• AGA 3 - 1992</li> <li>• ISO - 5167</li> <li>• AGA 3 - 2013</li> <li>• AGA 3 - 1992 (Errata 2013)</li> </ul>                                 |
|        | CalcVcone<br>Calc_Vcone       | Calculates various values associated with flow through a vcone type meter.                         | <ul style="list-style-type: none"> <li>• McCrometer</li> </ul>                                                                                                                                               |
|        | CalcMeter                     | Calculates various values associated with flow through diaphragm, rotary, and turbine meters.      | <ul style="list-style-type: none"> <li>• AGA 7 2006</li> <li>• Generic Rotary</li> <li>• Generic Turbine</li> <li>• GRI/SWRI Diaphragm</li> <li>• Rockwell Diaphragm</li> <li>• Sprague Diaphragm</li> </ul> |
|        | CalcQStd<br>Calc_QStd         | Calculates a standard volume based on specified flowing conditions.                                | <ul style="list-style-type: none"> <li>• AGA 7 2006</li> </ul>                                                                                                                                               |
|        | CalcPulse<br>Calc_Pulse       | Calculates a standard volume based on specified flowing conditions, meter factor, and pulse count. | <ul style="list-style-type: none"> <li>• AGA 7 2006</li> </ul>                                                                                                                                               |
|        | CalcUSM                       | Calculates various values associated with flow through an ultrasonic flow meter.                   | <ul style="list-style-type: none"> <li>• Daniels Meter</li> </ul>                                                                                                                                            |

| Module                   | Functions                             | Function Description                                                                                                          | Methods/Options                                                                                                                                                                                                                                                                                                                                                   |
|--------------------------|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Pipe                     | CalcPipeFlow<br>Calc_PipeFlow         | Calculates various values associated with flow through a pipe segment.                                                        |                                                                                                                                                                                                                                                                                                                                                                   |
|                          | CalcPipeEquivL<br>Calc_PipeEquivL     | Calculates the equivalent length in terms of an equivalent diameter based on a specified pipe or fitting diameter and length. |                                                                                                                                                                                                                                                                                                                                                                   |
|                          | CalcPipeVelocity<br>Calc_PipeVelocity | Calculates the maximum velocity associated with a volumetric flow rate, pipe diameter, and pressure.                          |                                                                                                                                                                                                                                                                                                                                                                   |
| Regulator & Relief Valve | CalcReg                               | Calculates various values associated with flow through a pressure regulator.                                                  | <ul style="list-style-type: none"> <li>• American</li> <li>• American AFV</li> <li>• Equimeter</li> <li>• Fisher Controls Universal</li> <li>• Grove 80</li> <li>• Grove 900</li> <li>• ISA S75.01</li> <li>• Mooney Controls Universal</li> <li>• Pietro Fiorentini</li> <li>• Rockwell</li> <li>• Table Based</li> </ul>                                        |
|                          | CalcRelief                            | Calculates various values associated with flow through a pressure regulator.                                                  | <ul style="list-style-type: none"> <li>• American</li> <li>• American AFV</li> <li>• API 520</li> <li>• ASME BPV</li> <li>• Equimeter</li> <li>• Fisher Controls Universal</li> <li>• Grove 80</li> <li>• Grove 900</li> <li>• ISA S75.01</li> <li>• Mooney Controls Universal</li> <li>• Pietro Fiorentini</li> <li>• Rockwell</li> <li>• Table Based</li> </ul> |

| Module                                                                                                                                                                              | Functions                           | Function Description                                                   | Methods/Options                                                                                                                                                                                                                                                                   |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------|------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Sonic                                                                                                                                                                               | CalcSonic<br>Calc_Sonic             | Calculates the speed of sound.                                         | <ul style="list-style-type: none"> <li>• AGA 8 2017, Detail &amp; Gross</li> <li>• AGA 8 2017, GERG - 2008</li> <li>• AGA 10 2003</li> <li>• Traditional Thermodynamic</li> </ul>                                                                                                 |
| Support                                                                                                                                                                             | CalcFact<br>Calc_Fact<br>CalcFact__ | Calculates various dimensional unit conversions.                       |                                                                                                                                                                                                                                                                                   |
|                                                                                                                                                                                     | CalcPatm<br>Calc_Patm               | Calculates atmospheric pressure values based on a specified elevation. | <ul style="list-style-type: none"> <li>• American Gas Association (AGA) - Measurement</li> <li>• Handbook of Chemistry &amp; Physics</li> <li>• International School of Hydrocarbon Measurement (ISHM)</li> <li>• National Oceanographic And Atmospheric (NOAA) - 1976</li> </ul> |
|                                                                                                                                                                                     | CalcPave<br>Calc_Pave               | Calculates the average of a pair of pressure values.                   |                                                                                                                                                                                                                                                                                   |
|                                                                                                                                                                                     | CalcT<br>Calc_T                     | Calculates various temperature conversion values.                      |                                                                                                                                                                                                                                                                                   |
| <i>Note</i> - The upper function name represents the name as used by the standard “c” dll interface. The lower name represents the name as used by the ActiveX and .net interfaces. |                                     |                                                                        |                                                                                                                                                                                                                                                                                   |

## API Interface Methods

Access to the API is provided through several types of dynamic link library (dll) interfaces. The interface types are listed below:

| Interface Type | Description                                                                                                                                                       |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| gc50_calcs.dll | Standard “c” type dynamic link library. Access can be made using any development or application environment that can “declare” a function within its environment. |

The non-calculation routines need to be able to find and access the gc50\_calcs.dll file. This may require specific placement of the gc50\_calcs.dll file.

### **Function Names & Parameters**

---

The following documentation reflects function names and parameter definitions associated with the standard “c” dll. For other interface types, the function name is as shown in the previous Modules & Functions table.

All parameters for values used in the gc50\_apix interface are defined as the “variant” type. All parameters for values used in the gc50\_apinet interface are defined as the “object” type. All parameters for values used in the gc50\_calcs interface are defined as shown in the following documentation.

When making calls directly to the standard “c” interface, variables/parameters are defined as follows:

short = 16-bit (2 Byte) signed integer

long = 32-bit (4 Byte) signed integer

float = 32-bit (4 Byte) single precision floating point numeric

double = 64-bit (8 Byte) double precision floating point numeric

## CalcElev - Atmospheric Elevation Calculation

The CalcElev routine calculates the elevation value associated with a specified atmospheric pressure value.

### Function Syntax

```
CalcElev  
(  
    short calc_method,  
    float patm  
)
```

### Variable Definitions

**calc\_method = Atmospheric Pressure Calculation Method** - Specifies the method to use to calculate the atmospheric pressure value. Short integer. Valid values are:

- 0 = None
- 1 = ISHM - Modified (International School of Hydrocarbon Measurement)
- 2 = Handbook of Chemistry and Physics
- 3 = AGA - Measurement
- 5 = NOAA - 1976
- 6 = ISHM (International School of Hydrocarbon Measurement)
- 7 = ASHRAE - 2009
- 11 = None - Entered

**patm = Atmospheric Pressure** - Specifies the atmospheric pressure to determine the elevation for. Floating point number, greater than 0, expressed in Psia.

### Return Values

The function returns a floating point number representing the atmospheric elevation of the arithmetic average of the specified elevation values. Or returns...

- 98 = Unsupported calculation method
- 99 = if the associated module is not enabled or licensed



### Licensing

---

Calculation Module: Support  
Additional Required Modules: None

### Notes & Considerations

---

- The *SetLicense* function must be called at least once before attempting to execute this calculation routine.
- See the appropriate GASCalc Calculation Reference for documentation and references for the calculations performed by this function.



## CalcFact - Dimensional Unit Conversion

The CalcFact routine calculates the factor required to convert from the specified dimensional units to the base dimensional units used by the various calculation routines.

### Function Syntax

```
CalcFact
(
    char *unit_group,
    char *unit_value,
    float hv
)
```

**Note...** \* Indicates a variable passed by reference (address), all other values passed by value. Calculated values will be returned in the associated “by reference” variables.

### Variable Definitions

**hv = Heating Value** - When converting flow rates in energy units to or from volumetric flow units, the heating value per unit volume must be passed to the conversion routine. Single precision floating point number. The value must be greater than 0 expressed in Btu/Cf if required, or 0 if not required.

**unit\_group = Unit Group** - Specifies the dimensional units group to use for the conversion calculation. Pointer to a character (string) value. Valid values are:

- “DUS” = Diameter group
- “ENUS” = Energy group
- “EUS” = Efficiency group
- “FUS” = Fuel group
- “HVUS” = Heating Value group
- “LUS” = Length group
- “PUS” = Pressure group
- “QUS” = Volumetric Flow group
- “UUS” = Viscosity group
- “VELUS” = Velocity group
- “XUS” = Coefficient of Thermal Expansion group

**unit\_value** = **Unit Value** - Specifies the dimensional units value to calculate the conversion factor for. Pointer to a character (string) value. Valid values are:

| Unit Group           | Valid Values        | Converts To |
|----------------------|---------------------|-------------|
| DUS - Diameter       | Centimeter, Cm      | Inches      |
|                      | Inches, Inch, In    |             |
|                      | Feet, Ft            |             |
|                      | Meters, Metres, M   |             |
|                      | Microns             |             |
|                      | Millimeter, Mm      |             |
| ENUS - Energy        | Btu                 | Therms      |
|                      | Kilowatt-Hour, Kwh  |             |
|                      | MBTU                |             |
|                      | MMBtu               |             |
|                      | MWh                 |             |
|                      | Therms              |             |
| EUS - Efficiency     | Decimal             | Decimal     |
|                      | Percent, %          |             |
| FUS - Fuel           | Cfh/Hp              | Cfh/Hp      |
|                      | Kw/Hp               |             |
|                      | Mcfh/Hp             |             |
| HVUS - Heating Value | Btu/Cf              | Btu/Cf      |
|                      | Btu/M^3             |             |
|                      | Joules/Cf           |             |
|                      | Joules/M^3          |             |
|                      | KJoules/Cf, KJ/Cf   |             |
|                      | KJoules/M^3, KJ/M^3 |             |

| Unit Group     | Valid Values                | Converts To |
|----------------|-----------------------------|-------------|
|                | MBtu/Cf                     |             |
|                | MBtu/M^3                    |             |
|                | MJoules/M^3                 |             |
| LUS - Length   | Centimeters, Centimeter, Cm | Feet        |
|                | Inches, Inch, In            |             |
|                | Feet, Ft                    |             |
|                | Kilometers, Kilometer, Km   |             |
|                | Meters, Metres, M           |             |
|                | Miles, Mile, Mi             |             |
| PUS - Pressure | Atmospheres, Atm            | Psi         |
|                | Bar                         |             |
|                | Feet H2O, Ft-H2O            |             |
|                | Inches H2O, In-WC           |             |
|                | Inches HG, In-Hg            |             |
|                | Kg/Cm^2                     |             |
|                | Kg/M^2                      |             |
|                | KPa                         |             |
|                | Millibar, MBar              |             |
|                | MM-H2O                      |             |
|                | MM-Hg                       |             |
|                | MPa                         |             |
|                | Osi                         |             |
|                | Pascals, Pascal             |             |
|                | Psf                         |             |
|                | Psi, Psig, Psia             |             |

| Unit Group            | Valid Values            | Converts To |
|-----------------------|-------------------------|-------------|
| QUS - Volumetric Flow | Btu/Hr ①                | Cfh         |
|                       | Ccf/Mo                  |             |
|                       | Cfd                     |             |
|                       | Cfh                     |             |
|                       | Cfm                     |             |
|                       | GJoules/d ①             |             |
|                       | GJoules/Hr ①            |             |
|                       | Kilowatts, Kilowatt, KW |             |
|                       | KJoules/Hr, KJoule/Hr ① |             |
|                       | Liters/Min              |             |
|                       | M^3/hr, M^3/h, M3h      |             |
|                       | M3/d, M3d               |             |
|                       | MBtu/Hr ①               |             |
|                       | Mcf/Mo                  |             |
|                       | Mcfd                    |             |
|                       | Mcfh                    |             |
|                       | MJoules/Hr, MJoule/Hr ① |             |
|                       | MM3d                    |             |
|                       | MMBtu/d                 |             |
|                       | MMBtu/Hr                |             |
|                       | MMcfd                   |             |
|                       | MMcfh                   |             |
|                       | MMM3/d, MMM3d           |             |
|                       | TJ/d                    |             |
|                       | TJ/h                    |             |

| Unit Group                              | Valid Values                | Converts To |
|-----------------------------------------|-----------------------------|-------------|
| UUS - Viscosity                         | Centipoise                  | Lbm/Ft-Sec  |
|                                         | Lbf -Sec/Ft^2               |             |
|                                         | Lbm/Ft-Sec                  |             |
|                                         | Micropoise                  |             |
|                                         | Pascal-Sec                  |             |
| VELUS - Velocity                        | Feet/Sec, Fps               | Ft/Sec      |
|                                         | Kilometers/Hr, Kph          |             |
|                                         | Meters/Sec, Metres/Sec, Mps |             |
|                                         | Mph                         |             |
| XUS - Coefficient of Thermal Expansion  |                             |             |
| ① Requires heating value to be defined. |                             |             |

## Return Values

The function returns a single precision floating point number, which represents the multiplication factor to convert from a value in unit\_value units to a value in the base dimensional units associated with the specified unit group. Or...

- 1 = Invalid unit\_group or unit\_value value.
- 99 = Module is not enabled or licensed.

## Licensing

Calculation Module: Support  
Additional Required Modules: None

## Notes & Considerations

- Temperature Values must be converted using the *CalcT* function.

- To convert a value in the passed (non-base) units to a value in base units, multiply the non-base value by the returned value. To convert a value in base units to a value in non-base units, divide by the returned value.
- The *SetLicense* function must be called at least once before attempting to execute this calculation routine.
- Use the CalcFact\_\_ (double underscore) function to return the conversion factor in double precision. All inputs are the same as the CalcFact function.
- See the GASCalc Calculation Reference for the Unit Conversion calculation for the unit conversion factors used by this function.

## CalcGP - Gas Properties Calculation

The CalcGP routine calculates a specified physical property of a specified gas composition for a specified set of base conditions.

### Function Syntax

```
CalcGP
(
    short property_type,
    short calc_method,
    float pb,
    float tb,
    short z_method,
    float *pp
)
```

**Note...** \* Indicates a variable passed by reference (address), all other values passed by value. Calculated values will be returned in the associated “by reference” variables.

### Variable Definitions

**calc\_method = Calculation Method** - Specifies the method to use to calculate the specified property\_type. Short integer. Valid values are dependent on the value of the property\_type variable, valid values are:

| Property Type Value            | Valid Values of Calculation Method                                                                                                                                                                |
|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0 - Volumetric Heating Value   | 0 = AGA 8 Method<br>1 = GPA 2172-86 Method<br>2 = GPA 2172-09 Method<br>3 = GPA 2172-09 C+ Method 1<br>4 = GPA 2172-09 C+ Method 2<br>5 = GPSA Method<br>6 = International Standard ISO 6976 2016 |
| 1 - Ideal Gas Specific Gravity | 0 = Standard GASCalc Method                                                                                                                                                                       |
| 2 - Real Gas Specific Gravity  | 0 = Standard GASCalc Method                                                                                                                                                                       |

| Property Type Value               | Valid Values of Calculation Method                  |
|-----------------------------------|-----------------------------------------------------|
| 3 - Ratio of Specific Heats       | 0 = Gas Engineers Handbook Method<br>1 = GPA Method |
| 4 - Dynamic or Absolute Viscosity | 0 = Standard GASCalc Method                         |
| 5 - Speed of Sound                | 0 = GPSA (Traditional Thermodynamic) Method         |

**pb = Base Pressure** - Specifies the pressure basis for stated volumetric flow rates (basis for a standard cubic foot). Single precision floating point number greater than 0 expressed in Psia.

**pp = Gas Components** - Specifies the mole fraction of the various components in the gas mixture. A pointer to single precision floating point number array. The array must be dimensioned to accommodate at least 24 components. Individual component values must be between 0.0 and 1.0. See the [CalcZF](#) function description for a list of the component order/assignment. Or...

When using Method 5 (Speed of Sound calculation):  
 pp[0] = Ratio of Specific Heats, dimensionless  
 pp[1] = Flowing Temperature, Rankine  
 pp[2] = Ideal Specific Gravity, dimensionless

**property type = Property Type** - Specifies the property type to be calculated. Short integer. Valid values are:

0 = Volumetric Heating Value  
 1 = Ideal Gas Specific Gravity  
 2 = Real Gas Specific Gravity  
 3 = Ratio of Specific Heats  
 4 = Dynamic or Absolute Viscosity  
 5 = Speed of Sound

**tb = Base Temperature** - Specifies the temperature basis for stated volumetric flow rates (basis for a standard cubic foot). Single precision floating point number greater than 0 expressed in Rankine.

**z\_method = Compressibility Calculation Method** - Specifies the method to use to calculate the compressibility factor. Short integer. Only used for property\_type value equal 2 (real gas specific gravity). Valid values are:

1 = AGA 8 1992 Detailed Method  
 10 = GERG 88 Detailed Method



---

**Return Values**

---

The function returns a single precision number indicating the calculated property\_type value as shown in the following table:

| Property Type                     | Return Value                                                   |
|-----------------------------------|----------------------------------------------------------------|
| 0 - Volumetric Heating Value      | Volumetric Heating Value expressed in Btu/cf.                  |
| 1 - Ideal Gas Specific Gravity    | Ideal Gas Specific Gravity value expressed as a decimal value. |
| 2 - Real Gas Specific Gravity     | Real Gas Specific Gravity value expressed as a decimal value.  |
| 3 - Ratio of Specific Heats       | Ratio of Specific Heats value expressed as a decimal value.    |
| 4 - Dynamic or Absolute Viscosity | Dynamic or Absolute Viscosity value expressed in Lbm/Ft-Sec.   |
| 5 - Speed of Sound                | Speed of Sound value expressed in Ft/Sec.                      |

Or if an error occurs, the following values are returned...

- 1 = Unknown calculation error.
- 2 = Invalid base compressibility value.
- 99 = Module is not enabled or licensed.
- 100 = Temperature out-of-range.
- 101 = Pressure out-of-range.

---

**Licensing**

---

Calculation Module: Gas Properties

Additional Required Modules: Compressibility\*

\*Only required if the real gas specific gravity values are to be calculated.

## Notes & Considerations

---

- All passed values must be expressed in the listed dimensional units for the associated variable. Calculated values are returned in the listed dimensional units. The *CalcFact* or *CalcT* functions can be used to obtain the required conversion factor for non-standard dimensional units.
- With the exception of the base pressure value, where passed pressure values are expressed in absolute pressure (Psia) dimensional units, the atmospheric pressure value must be added to the gauge pressure value (Psig) before passing the variable to the calculation function. Conversely, where a returned (calculated) value is expressed in absolute pressure units, the atmospheric pressure must be subtracted from the returned value to obtain the gauge pressure value. The atmospheric pressure value may be calculated using the *CalcPatm* function.
- The general process for using this function is as follows:
  - Collect the required variable values from User-entered values.
  - Convert the entered variable values to the appropriate dimensional units.
  - Call the calculation function.
  - If no error code is returned, use or display the returned value to the User. If an error code is returned, display an appropriate message to the User.
  - Store, print, or use the input values and results as appropriate.
- The *SetLicense* function must be called at least once before attempting to execute this calculation routine.
- See the appropriate GASCalc Calculation Reference for documentation and references for the calculations performed by this function.



## CalcLoss - Gas Loss Calculation

The CalcLoss routine calculates the volumetric flow rate from a damaged pipe. It returns the flow rate and pressure at the puncture location and the flow through the pipes upstream of (supplying) the damage location. The pipe damage can be specified as a puncture (opening in the sidewall of the pipe) or severed (completely cut in two). The function allows either a single (one-way) feed or a dual (two-way) feed to the damaged area.

### Function Syntax

```
CalcLoss
(
    short puncture,
    short eq_code,
    short pa_method,
    short z_method,
    short pipe2_ok,
    short sonic_pipe,

    float d1,
    float r1,
    float e1,
    float l1,
    double *q1,
    double max_1,

    float d2,
    float r2,
    float e2,
    float l2,
    double *q2,
    double max_2,

    float d3,
    float shape_x,
    double *q3,

    float p1,
    float t1,
    float z1,
    float *patm1,
```

float p2,  
float t2,  
float z2,  
float \*patm2,

float \*p3,  
float t3,  
float z3,  
float \*patm2,

float pb,  
float tb,

float sg,  
float mu,  
float cpcv

)

## Note...

\* Indicates a variable passed by reference (address), all other values passed by value. Calculated values will be returned in the associated “by reference” variables.

## Variable Definitions

**cpcv = Specific Heat Ratio** - Specifies the ratio of the average specific heat at constant pressure to the average specific heat at constant volume of the gas. Single precision floating point number greater than 0 expressed in Decimal.

**d1 = Supply 1 Diameter** - Specifies the equivalent inside diameter of the Supply 1 piping. Single precision floating point number greater than 0 expressed in Inches.

**e1 = Supply 1 Efficiency** - Specifies the pipe flow efficiency of the Supply 1 piping. Single precision floating point number greater than 0 expressed in Decimal.

**eq\_code = Pipe Equation Code** - Specifies which pipe flow equation to use. Short integer. See the [CalcPipeFlow](#) function description for valid values.

**l1 = Supply 1 Length** - Specifies the total equivalent hydraulic length of the Supply 1 piping. Single precision floating point number greater than 0 expressed in Feet.



**max\_1 = Supply 1 Flow Limit** - Specifies the limiting (maximum) flow allowed through the Supply 1 piping, at the specified base pressure and temperature. Double precision floating point number expressed in Cfh. Specify 0 for no limit, specify a value greater than 0 to limit the flow to that value.

**mu = Viscosity** - Specifies the average absolute or dynamic viscosity of the gas. Single precision floating point number greater than 0 expressed in Lbm/Ft-Sec.

**p1 = Supply 1 Pressure** - Specifies the supply (inlet) pressure of the Supply 1 piping. Single precision floating point number greater than one negative atmosphere expressed in Psig.

**p3 = Puncture Pressure** - Specifies the pressure at the puncture (just inside the pipe wall). A pointer to a single precision floating point number expressed in Psig. Calculated value.

**pa\_method = Atmospheric Pressure Calculation Method** - Specifies which method to use to calculate the atmospheric pressure value. Short integer. See the [CalcPatm](#) function description for valid values.

**patm1 = Supply 1 Atmospheric Pressure** - Specifies the calculated atmospheric pressure value at z1. A pointer to a single precision floating point number expressed in Psia. Calculated value.

**patm2 = Supply 2 Atmospheric Pressure** - Specifies the calculated atmospheric pressure value at z2. A pointer to a single precision floating point number expressed in Psia. Calculated value.

**patm3 = Puncture Atmospheric Pressure** - Specifies the calculated atmospheric pressure value at the elevation at the damage location (z3). A pointer to a single precision floating point number expressed in Psia. Calculated value.

**pb = Base Pressure** - Specifies the pressure basis for stated volumetric flow rates (basis for a standard cubic foot). Single precision floating point number greater than 0 expressed in Psia.

**pipe2\_ok = Two-Way Feed Flag** - Indicates whether one or two supply paths feed to the damaged area. Short integer. Valid values are:

- 0 = One supply path
- 1 = Two supply paths

**puncture = Damage Type** - Specifies whether severed or punctured. Short integer. Valid values are:

- 0 = Severed
- 1 = Punctured

**q1 = Supply 1 Flow Rate** - Specifies the flow through the Supply 1 piping feeding the puncture. A pointer to a double precision floating point number expressed in Cfh at the specified base pressure and temperature. Calculated value.



**q2 = Supply 2 Flow Rate** - Specifies the flow through the Supply 2 piping feeding the puncture. A pointer to a double precision floating point number expressed in Cfh at the specified base pressure and temperature. Calculated value.

**q3 = Puncture Flow Rate** - Specifies the flow through the puncture. A pointer to a double precision floating point number expressed in Cfh at the specified base pressure and temperature. Calculated value.

**r1 = Supply 1 Wall Roughness** - Specifies the pipe inside wall roughness of the Supply 1 piping. Only used for some pipe flow equations. Single precision floating point number greater than or equal to 0 expressed in Inches, when used - 0 when not used.

**sg = Specific Gravity** - Specifies the average specific gravity of the gas (ideal value compared to dry air). Single precision floating point number greater than 0 expressed in Decimal.

**sonic\_pipe = Sonic Velocity Limit Flag** - Indicates whether the flow velocity in the pipe segments is to be limited to sonic velocity or less. Short integer. Valid values are:

0 = Unrestricted  
1 = Restricted

**t1 = Supply 1 Temperature** - Specifies the average flowing gas temperature of the Supply 1 piping. Single precision floating point number greater than 0 expressed in Rankine.

**tb = Base Temperature** - Specifies the temperature basis for stated volumetric flow rates (basis for a standard cubic foot). Single precision floating point number greater than 0 expressed in Rankine.

**z1 = Supply 1 Elevation** - Specifies the height above sea level of the inlet end of the Supply 1 piping. Single precision floating point number expressed in Feet. Any valid elevation value.

**z\_method = Compressibility Calculation Method** - Specifies which method to use to calculate the compressibility factor. Valid values are:

0 = None

*Note* - The following can be set to 0 if only one supply.

**d2 = Supply 1 Diameter** - Specifies the equivalent inside diameter of the Supply 2 piping. Single precision floating point number greater than 0 expressed in Inches.

**e2 = Supply 2 Efficiency** - Specifies the pipe flow efficiency of the Supply 2 piping. Single precision floating point number greater than 0 expressed in Decimal.



**l2 = Supply 2 Length** - Specifies the total equivalent hydraulic length of the Supply 2 piping. Single precision floating point number greater than 0 expressed in Feet.

**max\_2 = Supply 2 Flow Limit** - Specifies the limiting (maximum) flow allowed through the Supply 2 piping. Double precision floating point number expressed in Cfh at the specified base pressure and temperature. Specify 0 for no limit, specify a value greater than 0 to limit the flow to that value.

**r2 = Supply 2 Wall Roughness** - Specifies the pipe inside wall roughness of the Supply 2 piping. Only used for some pipe flow equations. Single precision floating point number greater than or equal to 0 expressed in Inches, when used - 0 when not used.

*Note* - The following can be set to 0 if severed (not punctured).

**d3 = Puncture Diameter** - Specifies the equivalent inside diameter of the puncture. Single precision floating point number greater than 0 expressed in Inches.

**shape\_x = Shape Factor** - If the damage is a puncture, this item linearly adjusted the flow through the puncture. Floating point number greater than 0, normally 1.0.

*Note* - The following can be set to 0 if only one supply.

**p2 = Supply 2 Pressure** - Specifies the supply (inlet) pressure of the Supply 2 piping. Single precision floating point number greater than one negative atmosphere expressed in Psig.

**t2 = Supply 2 Temperature** - Specifies the average flowing gas temperature of the Supply 2 piping. Single precision floating point number greater than 0 expressed in Rankine.

**z2 = Supply 2 Elevation** - Specifies the height above sea level of the inlet end of the Supply 2 piping. Single precision floating point number expressed in Feet. Any valid elevation value.

*Note* - The following can be set to 0 if the damage type is severed, not punctured.

**t3 = Puncture Temperature** - Specifies the average flowing gas temperature to the puncture. Single precision floating point number greater than 0 expressed in Rankine.

**z3 = Puncture Elevation** - Specifies the height above sea level of the puncture. Single precision floating point number expressed in Feet. Any valid elevation value.

---

## Return Values

---

The function returns a short integer number indicating the calculation status. Possible return values are as follows:

- 1 = Successful calculation
- 0 = Not successful - unknown error
- 1 = Error calculating flow rate along Supply 1 piping
- 2 = Error calculating flow rate along Supply 2 piping
- 3 = Error calculating flow rate through the puncture
- 99 = Module not enabled or licensed

---

## Licensing

---

Calculation Module: Gas Loss

Additional Required Modules: Miscellaneous, Pipe Flow

---

## Notes & Considerations

---

- All values must be passed in listed dimensional units for the associated variable. Calculated values are returned in the listed dimensional units. The *CalcFact* or *CalcT* functions can be used to obtain the required conversion factor for non-standard dimensional units.
- If a series of pipes and fittings are used to define the Supply 1 and Supply 2 flow paths, the various components must be combined into a single pipe segment of equivalent hydraulic diameter and length. The *CalcPipeEquivL* function can be used to calculate the equivalent hydraulic length for a known pipe diameter and given equivalent hydraulic diameter. The equivalent hydraulic length of the individual components can be summed to determine the total equivalent hydraulic length for the flow path.
- If the damage type is a puncture (opening in the pipe sidewall), the damage can be specified as a circular hole, rectangular shape, or area. In the case of a non-circular hole, the puncture area must be converted to an equivalent hole diameter. The *CalcPuncture* function can be used to calculate the equivalent puncture diameter.
- With the exception of the base pressure value, where passed pressure values are expressed in absolute pressure (Psia) dimensional units, the atmospheric pressure value must be added to the gauge pressure value (Psig) before passing the variable to the calculation function. Conversely, where a returned (calculated) value is expressed in absolute pressure units, the atmospheric pressure must be subtracted from the returned value to obtain the gauge pressure value. The atmospheric pressure value may be calculated using the *CalcPatm* function.



- The general process for using this function is as follows:
  - Collect the required variable values from User-entered values.
  - Convert the entered variable values to the appropriate dimensional units.
  - Combine the supply path pipe and fitting components into a single equivalent pipe segment.
  - If the damage is a sidewall puncture, calculate the equivalent puncture diameter.
  - Call the calculation function.
  - If no error code is returned, convert the returned values to the appropriate dimensional units for display to the User. If an error code is returned, display an appropriate message to the User.
  - Store, print, or use the input values and results as appropriate.
- The *SetLicense* function must be called at least once before attempting to execute this calculation routine.
- See the appropriate GASCalc Calculation Reference for documentation and references for the calculations performed by this function and for additional considerations for calculating gas loss from a damaged pipe.

## CalcMeter - Displacement Meter Calculation

The CalcMeter routine calculates various values associated with the gas flow through a diaphragm, rotary, and turbine-type flow measurement meters.

### Function Syntax

```
CalcMeter
(
    short calc_method,
    short calc_var,

    float *pf,
    float *dp,
    double *qf,
    float tf,
    float sg,
    float zf,

    double rated_qf,
    float rated_pf,
    float rated_dp,
    float hf,
    float rated_tf,
    float rated_sg,
    float rated_zf,

    float meterfactor,

    short *detail_cnt,
    char *detail
)
```

**Note...** \* Indicates a variable passed by reference (address), all other values passed by value. Calculated values will be returned in the associated “by reference” variables.

## Variable Definitions

**calc\_method = Calculation Method** - Specifies the method to use to calculate the indicated variable. Short integer. Valid values are:

101 = Sprague Diaphragm Method  
102 = GRI/SWRI Diaphragm Method  
103 = Rockwell Diaphragm Method

201 = Generic Rotary Method

301 = Generic Turbine Method

**calc\_var = Variable Calculation Flag** - Specifies which variable is to be calculated. Short integer. Valid values are:

0 = Meter Pressure  
1 = Differential Pressure  
2 = Volumetric Flow Rate

**detail = Calculation Details** - Lists various intermediate and final calculation results. A pointer to a character (string) value at least 1280 characters in length. The format of the list is as follows:

Parameter 1 Name or Description  
Parameter 1 Value  
Value 1 Dimensional Units (if applicable)  
.  
.  
.  
Parameter detail\_cnt Name or Description  
Parameter detail\_cnt Value  
Value detail\_cnt Dimensional Units (if applicable)

Each item in the list ends with a “new line” character. (ASCII Character 10)

**detail\_cnt** = Number of parameter entries in the Calculation Detail list. A pointer to a short integer. Pass as -1 to suppress generation of the calculation details.

**dp = Differential Pressure** - Specifies the pressure differential across the meter at flowing conditions. A pointer to a single precision floating point number expressed in Psia. This variable can be specified (passed) or calculated. If specified, the value must be greater than 0. Pass as 0.0 when using method 201 and 301.



**hf = Frictional Pressure Loss** - Specifies the frictional pressure loss across the meter. A single precision floating point number, greater than or equal to 0, expressed in Psia. Pass 0.0 if unknown or not used. Pass as 0.0 when using method 201 and 301.

**meterfactor = Meter Correction Factor** - Specifies the meter correction factor. Single precision floating point number greater than 0. Pass 1.0 if unknown or not used.

**pf = Pressure** - Specifies the meter pressure value at flowing conditions. A pointer to a single precision floating point number expressed in Psia. This variable can be specified (passed) or calculated. If specified, the value must be greater than 0.

**rated\_pf = Pressure** - Specifies the meter pressure value used to establish the rated capacity value. Single precision floating point number greater than 0 expressed in Psia.

**qf = Flow Rate** - Specifies the volumetric flow rate through the meter at the flowing conditions. A pointer to a double precision floating point number expressed in Cfh at the rated base pressure and temperature used to adjust the rated capacity value. This variable can be specified (passed) or calculated. If specified, the value must be greater than 0.

**rated\_dp = Rated Differential Pressure** - Specifies the pressure differential used to establish the rated capacity value. A pointer to a single precision floating point number, greater than 0, expressed in Psia. Pass as 0.0 when using method 201 and 301.

**rated\_pf = Rated Pressure** - Specifies the meter pressure value used to establish the rated capacity value. Single precision floating point number greater than 0 expressed in Psia.

**rated\_qf = Rated Capacity** - Specifies the volumetric capacity rating of the meter at the specified rating conditions. A double precision floating point number, greater than 0, expressed in Cfh. If the rated base pressure, temperature, and compressibility are different than the actual base conditions, the value must be adjusted as described on the following *Notes & Considerations* section before passing.

**rated\_sg = Rated Specific Gravity** - Specifies the specific gravity of the gas (compared to dry air) used to establish the rated capacity value. Single precision floating point number greater than 0 expressed in Decimal.

**rated\_tf = Rated Temperature** - Specifies the temperature of the gas used to establish the rated capacity value. Single precision floating point number greater than 0 expressed in Rankine.

**rated\_zf = Rated Compressibility** - Specifies the compressibility value of the gas used to establish the rated capacity value. Single precision floating point number greater than 0. Pass 1.0 if unknown or not used.

**sg = Specific Gravity** - Specifies the average specific gravity of the gas (compared to dry air). Single precision floating point number greater than 0 expressed in Decimal.



**tf = Temperature** - Specifies the average temperature of the gas at flowing conditions. Single precision floating point number greater than 0 expressed in Rankine.

**zf = Compressibility** - Specifies the compressibility value of the gas at flowing conditions. Single precision floating point number greater than 0. Pass 1.0 if unknown or not used.

### Return Values

---

The function returns a short integer number indicating the calculation status. Possible return values are as follows:

- 1 = Successful calculation.
- 0 = Not successful - unknown error.
  
- 1 = Invalid rated specific gravity value.
- 2 = Invalid rated temperature value.
- 3 = Invalid rated compressibility value.
- 4 = Invalid specific gravity value.
- 5 = Invalid temperature value.
- 6 = Invalid compressibility value.
- 7 = Invalid differential pressure value (less than frictional pressure loss).
- 8 = Invalid rated capacity value.
- 9 = Invalid rated differential pressure value.
- 10 = Invalid rated pressure value.
- 11 = Invalid meter factor value.
  
- 97 = Unsupported calculation (unknown) variable.
- 98 = Unsupported calculation method.
- 99 = Module not enabled or licensed.

### Licensing

---

Calculation Module: Meter  
Additional Required Modules: None

### Notes & Considerations

---

- All passed values must be expressed in the listed dimensional units for the associated variable. Calculated values are returned in the listed dimensional units. The *CalcFact* or *CalcT* functions can be used to obtain the required conversion factor for non-standard dimensional units.



- With the exception of the base pressure value, where passed pressure values are expressed in absolute pressure (Psia) dimensional units, the atmospheric pressure value must be added to the gauge pressure value (Psig) before passing the variable to the calculation function. Conversely, where a returned (calculated) value is expressed in absolute pressure units, the atmospheric pressure must be subtracted from the returned value to obtain the gauge pressure value. The atmospheric pressure value may be calculated using the *CalcPatm* function.
- If the actual base conditions are different than the base conditions used to establish the rated capacity, the rated capacity value must be adjusted to actual conditions before passing. Use the *CalcBase* function to compute the required value
- If compressibility values are required, use the *CalcZF* function to compute the required values.
- The general process for using this function is as follows:
  - Collect the required variable values from User-entered values.
  - Convert the entered variable values to the appropriate dimensional units.
  - Calculate compressibility values as appropriate.
  - Adjust the rated capacity value as appropriate.
  - Call the calculation function.
  - If no error code is returned, use or display the returned value to the User. If an error code is returned, display an appropriate message to the User.
  - Store, print, or use the input values and results as appropriate.
- The *SetLicense* function must be called at least once before attempting to execute this calculation routine.
- The flow rate (qf) is at the *rated* base pressure and temperature of the meter. Use the *CalcBase* function to convert to another set of base conditions.
- See the appropriate GASCalc Calculation Reference for documentation and references for the calculations performed by this function.

## CalcOrifice - Orifice Meter Calculation

The CalcOrifice routine calculates various values associated with the gas flow through an orifice-type flow measurement meter.

### Function Syntax

```
CalcOrifice
(
    short calc_method,
    short calc_var,
    float dr,
    float *orf,
    float *dp,
    double *qf,
    float pf,
    float tf,
    float pb,
    float tb,
    float patm,
    float sg,
    short sg_real,
    float mu,
    float cpcv,
    short z_method,
    float *pp,
    float tr_orf,
    float tr_tube,
    float x_orf,
    float x_tube,
    short p_tap,
    short dp_tap,
    short *detail_cnt,
    char *detail
)
```

**Note...**

\* Indicates a variable passed by reference (address), all other values passed by value. Calculated values will be returned in the associated “by reference” variables.

## Variable Definitions

**calc\_method = Calculation Method** - Specifies the method to use to calculate the indicated variable. Short integer. Valid values are:

- 1 = AGA 3 1985 Method
- 2 = AGA 3 1992 Method
- 3 = ISO 5167 Method
- 4 = AGA 3 2013
- 5 = AGA 3 1992 Errata 2013

**calc\_var = Variable Calculation Flag** - Specifies which variable is to be calculated. Short integer. Valid values are:

- 1 = Orifice Size
- 2 = Volumetric Flow Rate
- 3 = Differential Pressure

**cpcv = Specific Heat Ratio** - Specifies the ratio of the average specific heat at constant pressure to the average specific heat at constant volume of the gas. Single precision floating point number greater than 0 expressed in Decimal.

**detail = Calculation Details** - Lists various intermediate and final calculation results. A pointer to a character (string) value at least 1280 characters in length. The format of the list is as follows:

```
Parameter 1 Name or Description
Parameter 1 Value
Value 1 Dimensional Units (if applicable)
.
.
.
Parameter detail_cnt Name or Description
Parameter detail_cnt Value
Value detail_cnt Dimensional Units (if applicable)
```

Each item in the list ends with a “new line” character. (ASCII Character 10)

**detail\_cnt = Number of parameter entries in the Calculation Detail list.** A pointer to a short integer. Pass as -1 to suppress generation of the calculation details.

**dp = Differential Pressure** - Specifies the pressure differential across the orifice plate. A pointer to a single precision floating point number expressed in Inches H<sub>2</sub>O. This variable can be specified (passed) or calculated. If specified, the value must be greater than 0.



**dp\_tap = Differential Pressure Tap Location** - Specifies the location of the differential pressure (dp) tap. Short integer. Valid values are:

- 0 = Flange
- 1 = Pipe
- 2 = Flange Corner (Calculation method 3 only)

**dr = Meter Tube Diameter** - Specifies the inside diameter of the meter tube (pipe) at the specified reference temperature **tr\_tube**. Single precision floating point number greater than 0 expressed in Inches.

**mu = Viscosity** - Specifies the absolute or dynamic viscosity of the gas. Single precision floating point number greater than 0 expressed in Lbm/Ft-Sec.

**orf = Orifice Diameter** - Specifies the diameter of the orifice at the specified reference temperature **tr\_orf**. A pointer to a single precision floating point number expressed in Inches. This variable can be specified (passed) or calculated. If specified, the value must be greater than 0.

**p\_tap = Pressure Tap Location** - Specifies the location of the static pressure (pf) tap. Short integer. Valid values are:

- 3 = Upstream
- 4 = Downstream

**patm = Atmospheric Pressure** - Specifies the calculated atmospheric pressure value associated with pf. A single precision floating point number greater than or equal to 0 expressed in Psia. Only required for certain compressibility calculation methods, pass 0 when not required.

**pb = Base Pressure** - Specifies the pressure basis for stated volumetric flow rates (basis for a standard cubic foot). Single precision floating point number greater than 0 expressed in Psia.

**pf = Pressure** - Specifies the static pressure value. Single precision floating point number greater than 0 expressed in Psia.

**pp = Gas Components** - Specifies the mole fraction of the various components in the gas mixture. A pointer to single precision floating point number array. See the [CalcZF](#) function description for array requirements and definition. Or...

If using the compressibility calculation method (z\_method) 11:

- pp[0] = gas compressibility factor at flowing conditions (pf and tf).
- pp[1] = gas compressibility factor at base conditions (pb and tb).
- pp[2] = dry air only compressibility factor at base conditions (pb and tb).
- pp[3] = gas compressibility factor at standard conditions (ps and ts).

If using the compressibility calculation method (z\_method) 21:

pp[0] = Supercompressibility correction factor (Fpv)

**qf = Flow Rate** - Specifies the volumetric flow rate through the orifice. A pointer to a double precision floating point number expressed in Cfh at the specified base pressure and temperature. This variable can be specified (passed) or calculated. If specified, the value must be greater than 0.

**sg = Specific Gravity** - Specifies the average specific gravity of the gas (ideal or real value compared to dry air). Single precision floating point number greater than 0 expressed in Decimal. Only required for certain compressibility calculation methods, pass 0 when not required.

**sg\_real = Specific Gravity Type Flag** - Specifies the specific gravity type. Short integer. Valid values are:

0 = Ideal gas value

1 = Real gas value

**tb = Base Temperature** - Specifies the temperature basis for stated volumetric flow rates (basis for a standard cubic foot). Single precision floating point number greater than 0 expressed in Rankine.

**tf = Temperature** - Specifies the average temperature of the gas. Single precision floating point number greater than 0 expressed in Rankine.

**tr\_orf = Orifice Reference Temperature** - Specifies the reference temperature for the orfr variable value. Single precision floating point number expressed in Rankine. Any valid temperature value.

**tr\_tube = Meter Tube Reference Temperature** - Specifies the reference temperature for the dr variable value. Single precision floating point number expressed in Rankine. Any valid temperature value.

**x\_orf = Orifice Expansion Factor** - Specifies the coefficient of linear thermal expansion for the orifice plate material. Single precision floating point number expressed in In/In-F. Any valid expansion value.

**x\_tube = Meter Tube Expansion Factor** - Specifies the coefficient of linear thermal expansion for the meter tube material. Single precision floating point number expressed in In/In-F. Any valid expansion value.

**z\_method = Compressibility Calculation Method** - Specifies the method to use to calculate the compressibility factor. Short integer. Valid values are:

0 = None

11 = Entered compressibility values

21 = Entered Supercompressibility correction factor (Fpv)

Or other values associated with the various calculation methods supported by the *CalcZF* function. See the [CalcZF](#) function description for other acceptable values.



## Return Values

The function returns a short integer number indicating the calculation status. Possible return values are as follows:

- 1 = Successful calculation.
- 0 = Not successful - unknown error.
- 1 = Unsupported compressibility calculation method.
- 3 = Beta Ratio too small.
- 4 = Beta Ratio too large.
- 5 = Reynolds Number too small.
- 6 = Orifice Size too small.
- 7 = Error calculating flow rate solution.
- 8 = Unable to calculate unknown value - zero flow value.
- 9 = Error calculating orifice size solution.
- 10 = Error calculating differential pressure solution.
- 98 = Unsupported calculation method.
- 99 = Module not enabled or licensed.

### Note...

If the compressibility is calculated, other error codes may be returned if the compressibility calculation fails. See the [CalcZF](#) function description for additional possible return values.

## Licensing

Calculation Module: Meter

Additional Required Modules: Compressibility\*

\*Only required if the Compressibility Calculation Method to be used is other than Method 0, 11, or 21.

## Notes & Considerations

- All passed values must be expressed in the listed dimensional units for the associated variable. Calculated values are returned in the listed dimensional units. The *CalcFact* or *CalcT* functions can be used to obtain the required conversion factor for non-standard dimensional units.



- With the exception of the base pressure value, where passed pressure values are expressed in absolute pressure (Psia) dimensional units, the atmospheric pressure value must be added to the gauge pressure value (Psig) before passing the variable to the calculation function. Conversely, where a returned (calculated) value is expressed in absolute pressure units, the atmospheric pressure must be subtracted from the returned value to obtain the gauge pressure value. The atmospheric pressure value may be calculated using the *CalcPatm* function.
- The general process for using this function is as follows:
  - Collect the required variable values from User-entered values.
  - Convert the entered variable values to the appropriate dimensional units.
  - Call the calculation function.
  - If no error code is returned, use or display the returned value to the User. If an error code is returned, display an appropriate message to the User.
  - Store, print, or use the input values and results as appropriate.
- The *SetLicense* function must be called at least once before attempting to execute this calculation routine.
- See the appropriate GASCalc Calculation Reference for documentation and references for the calculations performed by this function.

## CalcPatm - Atmospheric Pressure Calculation

---

The CalcPatm routine calculates the atmospheric pressure value for an average of a specified pair of elevation values.

### Function Syntax

---

```
CalcPatm
(
    short calc_method,
    float z1,
    float z2
)
```

### Variable Definitions

---

**calc\_method = Atmospheric Pressure Calculation Method** - Specifies the method to use to calculate the atmospheric pressure value. Short integer. Valid values are:

- 0 = None
- 1 = ISHM - Modified (International School of Hydrocarbon Measurement)
- 2 = Handbook of Chemistry and Physics
- 3 = AGA - Measurement
- 5 = NOAA - 1976
- 6 = ISHM (International School of Hydrocarbon Measurement)
- 7 = ASHRAE - 2009
- 11 = None - Entered

**z1 = First Elevation** - Specifies the first elevation. Floating point number expressed in Feet. Any valid elevation value.

**z2 = Second Elevation** - Specifies the second elevation. Floating point number expressed in Feet. Any valid elevation value.

### Return Values

---

The function returns a floating point number representing the atmospheric elevation of the arithmetic average of the specified elevation values. Or returns...

- 0 = Ignored
- 98 = Unsupported calculation method
- 99 = If the associated module is not enabled or licensed.

### Licensing

---

Calculation Module: Support  
Additional Required Modules: None

### Notes & Considerations

---

- To calculate the elevation for a single elevation value, enter the same value for both elevation variables.
- The *SetLicense* function must be called at least once before attempting to execute this calculation routine.
- See the appropriate GASCalc Calculation Reference for documentation and references for the calculations performed by this function.



## CalcPave - Average Pressure Calculation

---

The CalcPave routine calculates the hydraulic average value of two pipe pressure values.

### Function Syntax

---

```
CalcPave  
(  
    float p1,  
    float p2  
)
```

### Variable Definitions

---

**p1 = First Pressure** - Specifies a pressure value. Single precision floating point number greater than 0 expressed in Psia.

**p2 = Second Pressure** - Specifies a pressure value. Single precision floating point number greater than 0 expressed in Psia.

### Return Values

---

The function returns a floating point number expressed in Psia upon success. Or...

0 = Unsuccessful calculation.  
-99 = Module not enabled or licensed.

### Licensing

---

Calculation Module: Support  
Additional Required Modules: None

### Notes & Considerations

---

- All passed values must be expressed in the listed dimensional units for the associated variable. Calculated values are returned in the listed dimensional units. The *CalcFact* or *CalcT* functions can be used to obtain the required conversion factor for non-standard dimensional units.



- With the exception of the base pressure value, where passed pressure values are expressed in absolute pressure (Psia) dimensional units, the atmospheric pressure value must be added to the gauge pressure value (Psig) before passing the variable to the calculation function. Conversely, where a returned (calculated) value is expressed in absolute pressure units, the atmospheric pressure must be subtracted from the returned value to obtain the gauge pressure value. The atmospheric pressure value may be calculated using the *CalcPatm* function.
- The *SetLicense* function must be called at least once before attempting to execute this calculation routine.
- See the appropriate GASCalc Calculation Reference for documentation and references for the calculations performed by this function.

## **CalcPipeEquivL - Equivalent Pipe Length Calculation**

---

The CalcPipeEquivL routine calculates the equivalent hydraulic length expressed in terms of a certain specified equivalent hydraulic diameter of a pipe, or fitting, of a different diameter and length.

### **Function Syntax**

---

```
CalcPipeEquivL  
(  
    float base_d,  
    float seg_d,  
    float seg_l,  
    short eq_code  
)
```

### **Variable Definitions**

---

**base\_d = Base (equivalent) Diameter** - Specifies the equivalent inside diameter to express the equivalent length in. Floating point number greater than 0.

**eq\_code = Equation Code** - Specifies which pipe flow equation to use. Short integer. See the [CalcPipeFlow](#) function description for valid values.

**seg\_l = Length** - Specifies the total hydraulic length of the pipe segment to be converted. Floating point number greater than 0.

**seg\_d = Diameter** - Specifies the inside diameter of the pipe segment to be converted. Floating point number greater than 0 expressed in the same units as the base\_d value.

### **Return Values**

---

The function returns a floating point number representing the total equivalent hydraulic length of the original pipe segment expressed in terms of the specified base\_d diameter and in the same dimensional units as seg\_l. Or returns -99 if the associated module is not enabled or licensed.

## Licensing

---

Calculation Module: Pipe Flow

Additional Required Modules: None

## Notes & Considerations

---

- It is not required that the passed values be expressed in any particular dimensional units. However, the passed diameter values must be expressed in the same units. The returned length value will be in the same units as the passed length units. Care must be taken when combining a series of pipes and fittings using the value returned by the function to ensure that all of the length units are the same for the combined segments.
- The *SetLicense* function must be called at least once before attempting to execute this calculation routine.
- See the appropriate GASCalc Calculation Reference for documentation and references for the calculations performed by this function.



## CalcPipeFlow - Pipe Flow Calculation

The CalcPipeFlow routine calculates various values associated with gas flow in a pipe segment.

### Function Syntax

```
CalcPipeFlow
(
    short eq_code,
    short calc_var,
    short calc_temp,
    short pa_method,
    short z_method,

    float *d,
    float *r,
    float *e,
    float *l,
    double *qf,

    float *p1,
    float *p1atm,
    float t1,
    float z1,

    float *p2,
    float *p2atm,
    float *t2,
    float z2,

    float pb,
    float tb,

    float sg,
    float mu,
    float cpcv,

    float *pp,

    float *zf,
    float *zb,
```

```
float max_vel,  
float *vel,  
  
short *d_method  
short sonic_pipe  
)
```

**Note...**

\* Indicates a variable passed by reference (address), all other values passed by value. Calculated values will be returned in the associated “by reference” variables.

### Variable Definitions

**calc\_temp = Temperature Calculation Method** - Specifies which method to use to calculate the value of temperature gain or loss along the pipe segment. Short integer. Valid values are:

0 = None

**calc\_var = Calculation Variable** - Specifies which variable to calculate the value of. Short integer. Valid values are:

0 = Inside Diameter  
1 = Efficiency  
2 = Length  
3 = Inlet Pressure  
4 = Outlet Pressure  
5 = Volumetric Flow Rate  
6 = Pipe Wall Roughness

**cpcv = Specific Heat Ratio** - Specifies the ratio of the average specific heat at constant pressure to the average specific heat at constant volume of the gas. Single precision floating point number greater than 0 expressed in Decimal.

**d = Inside Diameter** - Specifies the equivalent inside diameter of the pipe segment. A pointer to a single precision floating point number greater than 0 expressed in Inches. Entered or calculated value.



**d\_method = Diameter Calculation Method** - Specifies the method used to calculate the diameter value. Only used when the diameter is being calculated. A pointer to a short integer number expressed in decimal. Calculated value. Valid return values are:

0 = Diameter was calculated based on the pressure drop across the pipe segment (difference between the inlet and outlet pressures).

1 = Diameter was calculated based on the maximum allowable velocity.

**e = Efficiency** - Specifies the pipe flow efficiency of the pipe segment. A pointer to a single precision floating point number greater than 0 expressed in Decimal. Entered or calculated value.

**eq\_code = Pipe Equation Code** - Specifies which pipe flow equation to use. Short integer. Valid values are:

- 1 = IGT Improved
- 2 = Spitzglass Low Pressure
- 3 = Spitzglass High Pressure
- 4 = Weymouth
- 6 = Panhandle A
- 7 = Panhandle B
- 8 = AGA Turbulent
- 9 = AGA Partially Turbulent
- 10 = AGA Fully Turbulent\*
- 13 = Oliphant
- 14 = Colebrook-White\*
- 15 = Mueller High Pressure
- 16 = Pole Low Pressure
- 18 = Mueller Low Pressure
- 19 = IGE3 Medium Pressure
- 27 = IGE3 Low Pressure
- 32 = IMC Low Pressure
- 33 = IMC High Pressure
- 58 = Renouard Low Pressure

\*Requires valid pipe wall roughness value to be entered.

**l = Length** - Specifies the total equivalent hydraulic length of the pipe segment. A pointer to a single precision floating point number greater than 0 expressed in Feet. Entered or calculated value.

**max\_vel = Maximum Allowable Velocity** - Specifies the maximum allowable gas velocity. Only used when the diameter value is being calculated. A single precision floating point number greater than 0 expressed in feet/sec.

**mu = Viscosity** - Specifies the average absolute or dynamic viscosity of the gas. Single precision floating point number greater than 0 expressed in Lbm/Ft-Sec.



**p1 = Inlet Pressure** - Specifies the supply (inlet) pressure of the pipe segment. A pointer to a single precision floating point number greater than one negative atmosphere expressed in Psig. Entered or calculated value.

**p1atm = Inlet Atmospheric Pressure** - Specifies the calculated atmospheric pressure value at the Inlet Elevation of the pipe. A pointer to a single precision floating point number expressed in Psia. Entered or calculated value.

**p2 = Outlet Pressure** - Specifies the outlet pressure of the pipe segment. A pointer to a single precision floating point number greater than one negative atmosphere expressed in Psig. Entered or calculated value. When `calc_var=5` (volumetric flow rate) and `sonic_pipe=1`, if sonic velocity is reached, returns outlet pressure at sonic velocity.

**p2atm = Outlet Atmospheric Pressure** - Specifies the calculated atmospheric pressure value at the outlet elevation of the pipe segment. A pointer to a single precision floating point number expressed in Psia. Entered or calculated value.

**pa\_method = Atmospheric Pressure Calculation Method** - Specifies which method to use to calculate the atmospheric pressure value. Short integer. See the [CalcPatm](#) function description for valid values.

**pb = Base Pressure** - Specifies the pressure basis for the volumetric flow rate (basis for a standard cubic foot). Single precision floating point number greater than 0 expressed in Psia.

**pp = Gas Components** - Specifies the mole fraction of the various components in the gas mixture. A pointer to single precision floating point number array. See the [CalcZF](#) function description for array requirements and definition. Can be set to an empty array if compressibility is not being calculated.

**qf = Flow Rate** - Specifies the volumetric flow rate through the pipe segment. A pointer to a double precision floating point number expressed in Cfh at the specified base pressure and temperature. Entered or calculated value.

**r = Wall Roughness** - Specifies the pipe inside wall roughness of the pipe segment. Only used for some pipe flow equations. A pointer to a single precision floating point number greater than or equal to 0 expressed in Inches, when used - 0 when not used. Entered or calculated value.

**sg = Specific Gravity** - Specifies the average specific gravity of the gas (ideal gas value compared to dry air). Single precision floating point number greater than 0 expressed in Decimal.

**sonic\_pipe = Sonic Velocity Handling Flag** - Specifies whether flow velocity in the pipe segment should be limited to sonic velocity.

**t1 = Inlet Temperature** - Specifies the flowing gas temperature at the inlet end of the pipe segment. Single precision floating point number greater than 0 expressed in Rankine.



**t2 = Outlet Temperature** - Specifies the flowing gas temperature at the outlet end of the pipe segment. A pointer to a single precision floating point number greater than 0 expressed in Rankine. Entered, or calculated value when calc\_temp is not equal to 0.

**tb = Base Temperature** - Specifies the temperature basis for the volumetric flow rate (basis for a standard cubic foot). Single precision floating point number greater than 0 expressed in Rankine.

**vel = Calculated Velocity** - Specifies the maximum calculated velocity - the velocity at the outlet conditions of the pipe segment. A pointer to a single precision floating point number expressed in feet/sec. Calculated value.

**z1 = Inlet Elevation** - Specifies the height above sea level of the inlet end of the pipe segment. Single precision floating point number expressed in Feet. Any valid elevation value.

**z2 = Outlet Elevation** - Specifies the height above sea level of the outlet end of the pipe segment. Single precision floating point number expressed in Feet. Any valid elevation value.

**z\_method = Compressibility Calculation Method** - Specifies which method to use to calculate the compressibility factor. See the [CalcZF](#) function description for valid values.

**zb = Base Compressibility** - Specifies the calculated compressibility at the specified base conditions. A pointer to a single precision floating point number expressed in decimal. Calculated value, if z\_method not equal to 11, otherwise a passed value greater than 0. Entered or calculated value.

**zf = Measured Compressibility** - Specifies the calculated compressibility at the specified measured conditions. A pointer to a single precision floating point number expressed in decimal. Calculated value, if z\_method not equal to 11, otherwise a passed value greater than 0. Entered or calculated value.

## Return Values

---

The function returns a short integer number indicating the calculation status. Possible return values are as follows:

- 1 = Successful calculation.
- 0 = Not successful - unknown error.
- 1 = Invalid Diameter value.
- 2 = Invalid Length value.
- 3 = Invalid Elevation difference - difference greater than pipe length.
- 4 = Invalid Efficiency value.
- 5 = Invalid Roughness value.
- 6 = Invalid Pipe Flow Equation when calculating roughness value.
- 7 = Invalid Flow Rate value.
- 8 = Invalid Average Temperature value.



- 9 = Invalid Gas Property value.
- 10 = Invalid Pressure value.
- 11 = Invalid Equation Code value.
- 12 = Invalid Maximum Velocity value.
- 15 = Invalid Equation number.
- 16 = General Pipe calculation error.
- 20 = Invalid Outlet Pressure value, unable to calculate velocity.
- 99 = Module not enabled or licensed.

Additional error codes may be returned if the compressibility factor is calculated and an error occurs during its calculation. See the [CalcZF](#) function description for additional values.

### Licensing

---

Calculation Module: Pipe Flow

Additional Required Modules: Miscellaneous, Compressibility\*

\*Only required if the Compressibility Calculation Method to be used is other than Method 0, 11, or 21.

### Notes & Considerations

---

- All values must be passed in listed dimensional units for the associated variable. Calculated values are returned in the listed dimensional units. The *CalcFact* or *CalcT* functions can be used to obtain the required conversion factor for non-standard dimensional units.
- If a series of pipes and fittings are used to define the Supply 1 and Supply 2 flow paths, the various components must be combined into a single pipe segment of equivalent hydraulic diameter and length. The *CalcPipeEquivL* function can be used to calculate the equivalent hydraulic length for a known pipe diameter and given equivalent hydraulic diameter. The equivalent hydraulic length of the individual components can be summed to determine the total equivalent hydraulic length for the flow path.
- With the exception of the base pressure value, where passed pressure values are expressed in absolute pressure (Psia) dimensional units, the atmospheric pressure value must be added to the gauge pressure value (Psig) before passing the variable to the calculation function. Conversely, where a returned (calculated) value is expressed in absolute pressure units, the atmospheric pressure must be subtracted from the returned value to obtain the gauge pressure value. The atmospheric pressure value may be calculated using the *CalcPatm* function.



- The general process for using this function is as follows:
  - Collect the required variable values from User-entered values.
  - Convert the entered variable values to the appropriate dimensional units.
  - Call the calculation function.
  - If no error code is returned, convert the returned values to the appropriate dimensional units for display to the User. If an error code is returned, display an appropriate message to the User.
  - Store, print, or use the input values and results as appropriate.
- The *SetLicense* function must be called at least once before attempting to execute this calculation routine.
- If *sonic\_pipe* = 1, the calculation routine handles sonic (choked) flow as follows:
  - calc\_var = “D”** - The calculated diameter is not allowed to be less than the value that would produce sonic flow for the given conditions.
  - calc\_var = “P2”** - The calculated pressure at the downstream end of the pipe is not allowed to be less than the value that would create sonic flow for the given conditions.
  - calc\_var = “Q”** - The calculated flow value is not allowed to be greater than the value that would produce sonic flow at the downstream end of the pipe at the given conditions. The resulting pressure, at sonic flow, at the downstream end of the pipe is returned in parameter p2.
- See the appropriate GASCalc Calculation Reference for documentation and references for the calculations performed by this function.

## CalcPipeVelocity - Pipe Velocity Calculation

The CalcPipeVelocity routine calculates the gas velocity for a specified diameter and conditions.

### Function Syntax

```
CalcPipeVelocity  
(  
    float d,  
    float pf,  
    float tf,  
    float pb,  
    float tb,  
    double qf,  
    float zf,  
    float zb  
)
```

### Variable Definitions

**d = Inside Diameter** - Specifies the equivalent inside diameter of the pipe segment. A single precision floating point number greater than 0 expressed in Inches.

**pb = Base Pressure** - Specifies the pressure basis for the flow rate value. Single precision floating point number greater than 0 expressed in Psia.

**pf = Measured Pressure** - Specifies the pressure value at measured conditions. Single precision floating point number greater than 0 expressed in Psia.

**qf = Flow Rate** - Specifies the volumetric flow rate through the pipe segment. A double precision floating point number expressed in Cfh at the specified base pressure and temperature.

**tb = Base Temperature** - Specifies the temperature basis for the flow rate value. Single precision floating point number greater than 0 expressed in Rankine.

**tf = Measured Temperature** - Specifies the gas temperature at measured conditions. Single precision floating point number greater than 0 expressed in Rankine.

**zb = Base Compressibility** - Specifies the calculated compressibility at the specified base conditions. A pointer to a single precision floating point number greater than 0 expressed in decimal.



**zf = Measured Compressibility** - Specifies the calculated compressibility at the specified measured conditions. A pointer to a single precision floating point number greater than 0 expressed in decimal.

### Return Values

---

Upon success, the function returns a floating number representing the calculated velocity expressed in feet/sec. Or upon failure returns...

0 = Error calculating velocity.  
-99 = Module not enabled or licensed.

### Licensing

---

Calculation Module: Pipe Flow  
Additional Required Modules: None

### Notes & Considerations

---

- All passed values must be expressed in the listed dimensional units for the associated variable. Calculated values are returned in the listed dimensional units. The *CalcFact* or *CalcT* functions can be used to obtain the required conversion factor for non-standard dimensional units.
- With the exception of the base pressure value, where passed pressure values are expressed in absolute pressure (Psia) dimensional units, the atmospheric pressure value must be added to the gauge pressure value (Psig) before passing the variable to the calculation function. Conversely, where a returned (calculated) value is expressed in absolute pressure units, the atmospheric pressure must be subtracted from the returned value to obtain the gauge pressure value. The atmospheric pressure value may be calculated using the *CalcPatm* function.
- The *SetLicense* function must be called at least once before attempting to execute this calculation routine.
- See the appropriate GASCalc Calculation Reference for documentation and references for the calculations performed by this function.

## CalcPulse - Pulse Output Meter Calculation

The CalcPulse routine calculates various values associated with gas volumes measured using a pulse output meter.

### Function Syntax

```
CalcPulse
(
    short calc_method,
    short calc_var,
    float *k_factor,
    long *pulse_cnt,
    double *q,
    float pf,
    float tf,
    float pb,
    float tb,
    float patm,
    float sg,
    short z_method,
    float *pp,
    float *zf,
    float *zb
)
```

**Note...** \* Indicates a variable passed by reference (address), all other values passed by value. Calculated values will be returned in the associated “by reference” variables.

### Variable Definitions

**calc\_method = Calculation Method** - Specifies the method to use to calculate the indicated variable. Short integer. Valid values are:

0 = AGA 7 2006 Method



**calc\_var = Variable Calculation Flag** - Specifies which variable is to be calculated. Short integer. Valid values are:

- 0 = Meter Factor (k\_factor)
- 1 = Pulse Count (pulse\_cnt)
- 2 = Volume (q)

**k\_factor = Meter Factor** - Specifies the number of pulses per actual volume. Single precision floating point number expressed in pulses per Cf. This variable can be specified (passed) or calculated. If specified, the value must be greater than 0.

**patm = Atmospheric Pressure** - Specifies the calculated atmospheric pressure value associated with pf. A single precision floating point number greater than or equal to 0 expressed in Psia. Only required for certain compressibility calculation methods, pass 0 when not required.

**pb = Base Pressure** - Specifies the pressure basis for stated volumetric flow rates (basis for a standard cubic foot). Single precision floating point number greater than 0 expressed in Psia.

**pf = Pressure** - Specifies the static upstream pressure value. Single precision floating point number greater than 0 expressed in Psia.

**pp = Gas Components** - Specifies the mole fraction of the various components in the gas mixture. A pointer to single precision floating point number array. See the [CalcZF](#) function description for array requirements and definition. Or...

If using the compressibility calculation method (z\_method) 21:  
pp[0] = Supercompressibility correction factor (Fpv, zb/zf)

**pulse\_cnt = Pulse Count** - Specifies the number of pulses for the desired period. Long integer. This variable can be specified (passed) or calculated. If specified, the value must be greater than 0.

**q = Volume** - Specifies the volume corresponding to the pulse count. A pointer to a double precision floating point number expressed in Cfh at the specified base pressure and temperature. This variable can be specified (passed) or calculated. If specified, the value must be greater than 0.

**sg = Specific Gravity** - Specifies the average specific gravity of the gas (ideal or real value compared to dry air). Single precision floating point number greater than 0 expressed in Decimal. Only required for certain compressibility calculation methods, pass 0 when not required.

**tb = Base Temperature** - Specifies the temperature basis for stated volumetric flow rates (basis for a standard cubic foot). Single precision floating point number greater than 0 expressed in Rankine.

**tf = Temperature** - Specifies the average temperature of the gas. Single precision floating point number greater than 0 expressed in Rankine.



**z\_method = Compressibility Calculation Method** - Specifies the method to use to calculate the compressibility factor. Short integer. Valid values are:

- 0 = None
- 11 = Entered compressibility values
- 21 = Entered Supercompressibility correction factor (Fpv)

Or other values associated with the various calculation methods supported by the *CalcZF* function. See the [CalcZF](#) function description for other acceptable values.

**zb = Base Compressibility** - Specifies the calculated compressibility at the specified base conditions. A pointer to a single precision floating point number expressed in decimal. Calculated value, if z\_method not equal to 11 or 21, otherwise a passed value greater than 0.

**zf = Measured Compressibility** - Specifies the calculated compressibility at the specified measured conditions. A pointer to a single precision floating point number expressed in decimal. Calculated value, if z\_method not equal to 11 or 21, otherwise a passed value greater than 0.

### Return Values

The function returns a short integer number indicating the calculation status. Possible return values are as follows:

- 1 = Successful calculation.
- 0 = Not successful - unknown error.
- 1 = Invalid Volume value.
- 3 = Invalid Static Pressure value.
- 4 = Invalid Meter Factor.
- 99 = Module not enabled or licensed.
- 10X = Error calculating compressibility factor - see the [CalcZF](#) function description for error codes.

**Note...**

If the compressibility is calculated, other error codes may be returned if the compressibility calculation fails. See the [CalcZF](#) function description for additional possible return values.

## Licensing

---

Calculation Module: Meter

Additional Required Modules: Compressibility\*

\*Only required if the Compressibility Calculation Method to be used is other than Method 0, 11, or 21.

## Notes & Considerations

---

- All passed values must be expressed in the listed dimensional units for the associated variable. Calculated values are returned to the listed dimensional units. The *CalcFact* or *CalcT* functions can be used to obtain the required conversion factor for non-standard dimensional units.
- With the exception of the base pressure value, where passed pressure values are expressed in absolute pressure (Psia) dimensional units, the atmospheric pressure value must be added to the gauge pressure value (Psig) before passing the variable to the calculation function. Conversely, where a returned (calculated) value is expressed in absolute pressure units, the atmospheric pressure must be subtracted from the returned value to obtain the gauge pressure value. The atmospheric pressure value may be calculated using the *CalcPatm* function.
- The general process for using this function is as follows:
  - Collect the required variable values from User-entered values.
  - Convert the entered variable values to the appropriate dimensional units.
  - Call the calculation function.
  - If no error code is returned, use or display the returned value to the User. If an error code is returned, display an appropriate message to the User.
  - Store, print, or use the input values and results as appropriate.
- The *SetLicense* function must be called at least once before attempting to execute this calculation routine.

## CalcPuncture - Equivalent Puncture Diameter Calculation

The CalcPuncture routine calculates the equivalent hydraulic diameter of a puncture shape or area.

### Function Syntax

```
CalcPuncture
(
    short style,
    float x,
    float y
)
```

### Variable Definitions

**style = Puncture Style** - Specifies the puncture type. Short integer number. Valid values are:

0 = circular hole  
 1 = rectangular area  
 2 = irregular area

**x** = The definition of this item varies according to the value of the **style** variable. The item is defined as follows:

| Style Value | Definition                                                    | Allowable Values                                | Dimensional Units |
|-------------|---------------------------------------------------------------|-------------------------------------------------|-------------------|
| 0           | Diameter of a circular hole approximating the puncture shape. | Single precision floating point greater than 0. | Inches            |
| 1           | Width of a rectangular area approximating the puncture shape. | Single precision floating point greater than 0. | Inches            |
| 2           | Area of an irregular shape approximating the puncture shape.  | Single precision floating point greater than 0. | Square Inches     |

y = The definition of this item varies according to the value of the **style** variable. The item is defined as follows:

| Style Value | Definition                                                     | Allowable Values                                | Dimensional Units |
|-------------|----------------------------------------------------------------|-------------------------------------------------|-------------------|
| 0           | Not used.                                                      | Single precision floating point equal to 0.     | N/A               |
| 1           | Height of a rectangular area approximating the puncture shape. | Single precision floating point greater than 0. | Inches            |
| 2           | Not used.                                                      | Single precision floating point equal to 0.     | N/A               |

### Return Values

The function returns a floating point number representing the equivalent diameter of the puncture shape, expressed in Inches. Or returns -99 if the associated module is not enabled or licensed.

### Licensing

Calculation Module: Gas Loss  
Additional Required Modules: None

### Notes & Considerations

- The *SetLicense* function must be called at least once before attempting to execute this calculation routine.
- See the appropriate GASCalc Calculation Reference for documentation and references for the calculations performed by this function.

## CalcQBase - Base Volume Conversion

The CalcQBase routine converts a volume referenced to one set of base conditions to the equivalent volume referenced to another set of base conditions.

### Function Syntax

```
CalcQBase
(
    short calc_var,

    double *qb1,
    float pb1,
    float tp1,
    float zb1,

    double *qb2,
    float pb2,
    float tp2,
    float zb2,
)
```

**Note...**

\* Indicates a variable passed by reference (address), all other values passed by value. Calculated values will be returned in the associated “by reference” variables.

### Variable Definitions

**calc\_var = Calculation Variable** - Specifies which variable to calculate a value for. Short integer. Valid values are:

0 = Volume at base conditions 1  
1 = Volume at base conditions 2

**pb1 = Base Pressure At Condition 1** - Specifies the pressure basis for the volume value at the first set of conditions. Single precision floating point number greater than 0 expressed in a consistent dimensional unit.

**pb2 = Base Pressure At Condition 2** - Specifies the pressure basis for the volume value at the second set of conditions. Single precision floating point number greater than 0 expressed in a consistent dimensional unit.



**qb1 = Volume At Condition 1** - Specifies the volume or volumetric flow rate referenced to the first set of conditions. A pointer to a double precision floating point number expressed in a consistent dimensional unit. Entered or calculated value.

**qb2 = Volume At Condition 2** - Specifies the volume or volumetric flow rate referenced to the second set of conditions. A pointer to a double precision floating point number expressed in a consistent dimensional unit. Entered or calculated value.

**tb1 = Base Temperature At Condition 1** - Specifies the temperature basis for the volume value at the first set of conditions. Single precision floating point number greater than 0 expressed in a consistent dimensional unit.

**tb2 = Base Temperature At Condition 2** - Specifies the temperature basis for the volume value at the second set of conditions. Single precision floating point number greater than 0 expressed in a consistent dimensional unit.

**zb1 = Base Compressibility At Condition 1** - Specifies the calculated compressibility at the first set of conditions. Single precision floating point number greater than 0. Pass 1.0 if unknown.

**zb2 = Base Compressibility At Condition 2** - Specifies the calculated compressibility at the second set of conditions. Single precision floating point number greater than 0. Pass 1.0 if unknown.

### Return Values

---

The function returns a short integer indicating the success of the calculation.

- 1 = Successful calculation.
- 1 = Invalid base temperature value - condition 1.
- 2 = Invalid base temperature value - condition 2.
- 3 = Invalid base pressure value - condition 1.
- 4 = Invalid base pressure value - condition 2.
- 5 = Invalid base compressibility value - condition 1.
- 6 = Invalid base compressibility value - condition 2.
- 7 = Invalid volume value - condition 1.
- 8 = Invalid volume value - condition 2.
- 97 = Unsupported calculation (unknown) variable.
- 99 = Module not enabled or licensed.

## Licensing

---

Calculation Module: Meter Values  
Additional Required Modules: None

## Notes & Considerations

---

- All passed values must be expressed in the consistent dimensional units for the associated variable. Calculated values are returned in the dimensional units of the passed value. The *CalcFact* or *CalcT* functions can be used to obtain the required conversion factor for non-consistent dimensional units.
- If compressibility values are required, use the *CalcZF* function to compute the required values.
- The general process for using this function is as follows:
  - Collect the required variable values from User-entered values.
  - Convert the entered variable values to the appropriate dimensional units.
  - Call the calculation function.
  - If no error code is returned, use or display the returned value to the User. If an error code is returned, display an appropriate message to the User.
  - Store, print, or use the input values and results as appropriate.
- The *SetLicense* function must be called at least once before attempting to execute this calculation routine.
- See the appropriate GASCalc Calculation Reference for documentation and references for the calculations performed by this function.

## CalcQStd - Standard Volume Calculation

The CalcQStd routine calculates the standard (base) volume for a measured volume at a specified set of conditions and gas properties.

### Function Syntax

```
CalcQStd
(
    short calc_var,
    float pf,
    float tf,
    double *qf,
    float pb,
    float tb,
    double *qb,
    float patm,
    float sg,
    short z_method,
    float *pp,
    float *zf,
    float *zb
)
```

**Note...**

\* Indicates a variable passed by reference (address), all other values passed by value. Calculated values will be returned in the associated “by reference” variables.

### Variable Definitions

**calc\_var = Calculation Variable** - Specifies which variable to calculate a value for. Short integer. Valid values are:

- 0 = Measured Volume
- 1 = Standard/Base Volume

**patm = Atmospheric Pressure** - Specifies the calculated atmospheric pressure value associated with pf. A single precision floating point number greater than 0 expressed in Psia. Only required for certain compressibility calculation methods, pass 0 when not required.



**pb = Base Pressure** - Specifies the pressure basis for the standard/base volume value. Single precision floating point number greater than 0 expressed in Psia.

**pf = Measured Pressure** - Specifies the pressure value at measured conditions. Single precision floating point number greater than 0 expressed in Psia.

**pp = Gas Components** - Specifies the mole fraction of the various components in the gas mixture. A pointer to single precision floating point number array. See the [CalcZF](#) function description for array requirements and definition. Or...

If using the compressibility calculation method (z\_method) 21:

pp[0] = Supercompressibility correction factor (Fpv, zb/zf)

**qb = Volume At Base Conditions** - Specifies the volume or volumetric flow rate referenced to the base conditions. A pointer to a double precision floating point number expressed in a consistent dimensional unit. Entered or calculated value.

**qf = Volume At Flowing Conditions** - Specifies the volume or volumetric flow rate at the specified flowing conditions. A pointer to a double precision floating point number expressed in a consistent dimensional unit. Entered or calculated value.

**sg = Specific Gravity** - Specifies the average specific gravity of the gas (ideal value compared to dry air). Single precision floating point number greater than 0 expressed in Decimal. Only required for certain compressibility calculation methods, pass 0 when not required.

**tb = Base Temperature** - Specifies the temperature basis for the standard/base volume value. Single precision floating point number greater than 0 expressed in Rankine.

**tf = Measured Temperature** - Specifies the gas temperature at measured conditions. Single precision floating point number greater than 0 expressed in Rankine.

**z\_method = Compressibility Calculation Method** - Specifies the method to use to calculate the compressibility factor. Short integer. Valid values are:

0 = None

11 = Entered compressibility values - see the following *Notes & Considerations* section for required values.

*Note* - See the [CalcZF](#) function description for other acceptable values.

**zb = Base Compressibility** - Specifies the calculated compressibility at the specified base conditions. A pointer to a single precision floating point number expressed in decimal. Calculated value, if z\_method not equal to 11 or 21, otherwise a passed value greater than 0.



**zf = Measured Compressibility** - Specifies the calculated compressibility at the specified measured conditions. A pointer to a single precision floating point number expressed in decimal. Calculated value, if *z\_method* not equal to 11 or 21, otherwise a passed value greater than 0.

### Return Values

---

The function returns a short integer indicating the success of the calculation.

1 = Successful calculation.

-1 = Error calculating measured volume. Indicates an invalid Base Pressure, Base Temperature, or Measured Compressibility value.

-2 = Error calculating standard/base volume. Indicates an invalid Measured Pressure, Measured Temperature, or Base Compressibility value.

-10X = Error calculating compressibility factor - see the [CalcZF](#) function description for error codes.

### Licensing

---

Calculation Module: Meter Values

Additional Required Modules: Compressibility\*

\*Only required if the Compressibility Calculation Method to be used is other than Method 0, 11, or 21.

### Notes & Considerations

---

- All passed values must be expressed in the listed dimensional units for the associated variable. Calculated values are returned in the listed dimensional units. The *CalcFact* or *CalcT* functions can be used to obtain the required conversion factor for non-standard dimensional units.

- With the exception of the base pressure value, where passed pressure values are expressed in absolute pressure (Psia) dimensional units, the atmospheric pressure value must be added to the gauge pressure value (Psig) before passing the variable to the calculation function. Conversely, where a returned (calculated) value is expressed in absolute pressure units, the atmospheric pressure must be subtracted from the returned value to obtain the gauge pressure value. The atmospheric pressure value may be calculated using the *CalcPatm* function.

- The general process for using this function is as follows:
  - Collect the required variable values from User-entered values.
  - Convert the entered variable values to the appropriate dimensional units.
  - Call the calculation function.
  - If no error code is returned, use or display the returned value to the User. If an error code is returned, display an appropriate message to the User.
  - Store, print, or use the input values and results as appropriate.
- The *SetLicense* function must be called at least once before attempting to execute this calculation routine.
- See the appropriate GASCalc Calculation Reference for documentation and references for the calculations performed by this function.

## CalcReg - Regulator Flow Calculation

The CalcReg routine calculates various values associated with gas flow through a regulator.

### Function Syntax

```
CalcReg
(
    short calc_var,
    short eq_code,
    short eq2_code,

    float *cg,
    float parm_1,
    float parm_2,
    float parm_3,
    float parm_4,
    float min_dp,

    double *qf,
    double *p1,
    double *p2,
    float patm,
    float tf,

    float pb,
    float tb,

    float sg,
    float mu,
    float cpcv,
    float zf,

    short *flowmode,
    char *tablename
)
```

**Note...**

\* Indicates a variable passed by reference (address), all other values passed by value. Calculated values will be returned in the associated “by reference” variables.

## Variable Definitions

**calc\_var = Calculation Variable** - Specifies which variable to calculate the value of. Short integer. Valid values are:

- 1 = Valve Factor (cg)
- 2 = Volumetric Flow Rate
- 3 = Inlet Pressure
- 4 = Outlet Pressure

**cg = Valve Factor** - Specifies the gas valve factor for the regulator. A pointer to a single precision floating point number greater than 0. Entered or calculated value.

**cpcv = Specific Heat Ratio** - Specifies the ratio of the average specific heat at constant pressure to the average specific heat at constant volume of the gas. Single precision floating point number greater than 0 expressed in Decimal.

**eq\_code = Regulator Equation Code** - Specifies which regulator flow equation to use. Short integer. Valid values are:

- 0 = Fisher Controls
- 1 = Rockwell
- 2 = Equimeter
- 3 = American AFV
- 4 = American
- 5 = ISA S75.01
- 6 = Grove 900
- 7 = Mooney Controls
- 8 = Groove 80
- 9 = Table Interpolation
- 10 = Pietro Fiorentini

**eq2\_code = Secondary Regulator Equation Code** - When eq\_code=10, specifies which regulator flow equation to use if the entered values are outside the limit of the table. Short integer. Valid values are the same as eq\_code.

**flowmode = Flow Mode** - Specifies the type of flow occurring in the regulator. A pointer to a short integer number. Calculated value. Possible return values are:

- 1 = Sub-critical/Sub-sonic flow
- 2 = Critical/Sonic flow
- 3 = Zero flow



**min\_dp = Minimum Differential Pressure** - Specifies the minimum differential pressure required to fully open the valve. Single precision floating point number greater than 0 expressed in Psia.

**mu = Viscosity** - Specifies the average absolute or dynamic viscosity of the gas. Single precision floating point number greater than 0 expressed in Lbm/Ft-Sec.

**p1 = Inlet Pressure** - Specifies the upstream (inlet) pressure of the regulator. A pointer to a double precision floating point number greater than one negative atmosphere expressed in Psia. Entered or calculated value.

**p2 = Outlet Pressure** - Specifies the downstream (outlet) pressure of the regulator. A pointer to a double precision floating point number greater than one negative atmosphere expressed in Psia. Entered or calculated value.

**parm\_1 = Parameter 1** - Specifies the value of a particular parameter associated with the specified sizing equation. The definition of the parameter varies according to the specified equation. See the following *Notes & Considerations* section for information about the parameter definitions.

**parm\_2 = Parameter 2** - Specifies the value of a particular parameter associated with the specified sizing equation. The definition of the parameter varies according to the specified equation. See the following *Notes & Considerations* section for information about the parameter definitions.

**parm\_3 = Parameter 3** - Specifies the value of a particular parameter associated with the specified sizing equation. The definition of the parameter varies according to the specified equation. See the following *Notes & Considerations* section for information about the parameter definitions.

**parm\_4 = Parameter 4** - Specifies the value of a particular parameter associated with the specified sizing equation. The definition of the parameter varies according to the specified equation. See the following *Notes & Considerations* section for information about the parameter definitions.

**patm = Atmospheric Pressure** - Specifies the calculated atmospheric pressure value at the elevation of the regulator. A single precision floating point number expressed in Psia. Only used by equation code 9. Pass 0.0 when not used.

**pb = Base Pressure** - Specifies the pressure basis for the volumetric flow rate (basis for a standard cubic foot). Single precision floating point number greater than 0 expressed in Psia.

**qf = Flow Rate** - Specifies the volumetric flow rate through the regulator. A pointer to a double precision floating point number expressed in Cfh at the specified base pressure and temperature. Entered or calculated value.

**sg = Specific Gravity** - Specifies the average specific gravity of the gas (ideal gas value compared to dry air). Single precision floating point number greater than 0 expressed in Decimal.

**tf = Inlet Temperature** - Specifies the flowing gas temperature at the inlet to the regulator. Single precision floating point number greater than 0 expressed in Rankine.

**tablename = Capacity Table Name** - Specifies the name of the capacity table for the associated regulator. A pointer to a character string. Only used when the eq\_code value is 9. Pass an empty character string when not used.

**tb = Base Temperature** - Specifies the temperature basis for the volumetric flow rate (basis for a standard cubic foot). Single precision floating point number greater than 0 expressed in Rankine.

**zf = Compressibility** - Specifies the compressibility at the specified inlet conditions. A single precision floating point number, greater than 0, expressed in decimal. Pass 1.0 if unknown or unused. Not used by all equations.

### Return Values

---

The function returns a short integer number indicating the calculation status. Possible return values are as follows:

- 1 = Successful calculation.
- 0 = Not successful - unknown error.
  
- 1 = Invalid Valve Factor (Cg).
- 2 = Invalid Flow Rate value.
- 3 = Invalid Inlet Pressure value.
- 4 = Invalid Pressure value - the inlet pressure must be greater than the outlet pressure.
- 5 = Invalid Inlet Temperature value.
- 6 = Invalid Specific Gravity value.
- 7 = Invalid Specific Heat Ratio value.
- 8 = Invalid Compressibility value.
- 9 = Minimum Differential Pressure value not met.
- 10 = Out-of-range value.
- 11 = Specified table not found.
- 12 = Unsupported calculation variable.
- 13 = Cannot calculate Cg or Cv factor using table method.
- 14 = Invalid C1 value.
- 20 = Invalid table name.
  
- 97 = Unsupported calculation variable.
- 98 = Unsupported equation code.
- 99 = Module not enabled or licensed.

---

## Licensing

---

Calculation Module: Regulator & Relief Valve

Additional Required Modules: None

---

## Notes & Considerations

---

- All values must be passed in listed dimensional units for the associated variable. Calculated values are returned in the listed dimensional units. The *CalcFact* or *CalcT* functions can be used to obtain the required conversion factor for non-standard dimensional units.
- With the exception of the base pressure value, where passed pressure values are expressed in absolute pressure (Psia) dimensional units, the atmospheric pressure value must be added to the gauge pressure value (Psig) before passing the variable to the calculation function. Conversely, where a returned (calculated) value is expressed in absolute pressure units, the atmospheric pressure must be subtracted from the returned value to obtain the gauge pressure value. The atmospheric pressure value may be calculated using the *CalcPatm* function.
- The definition for the parameter values varies according to the equation used. See the GASCalc Regulator Calculation Reference for a description of the various definitions.
- The general process for using this function is as follows:
  - Collect the required variable values from User-entered values.
  - Convert the entered variable values to the appropriate dimensional units.
  - Call the calculation function.
  - If no error code is returned, convert the returned values to the appropriate dimensional units for display to the User. If an error code is returned, display an appropriate message to the User.
  - Store, print, or use the input values and results as appropriate.
- When using equation code 9, the specified capacity table must be in GASCalc 5.0 format, specifically in table version GC.5.1 format. The GASCalc software can be used to create the required table.
- The *SetLicense* function must be called at least once before attempting to execute this calculation routine.
- See the appropriate GASCalc Calculation Reference for documentation and references for the calculations performed by this function.

## CalcRelief - Relief Valve Flow Calculation

The CalcRelief routine calculates various values associated with gas flow through a pressure relief valve.

### Function Syntax

```
CalcRelief
(
    short calc_var,
    short eq_code,
    short eq2_code,

    float *area,
    float *cg,
    float parm_1,
    float parm_2,
    float parm_3,
    float parm_4,
    float min_dp,

    double *qf,
    double *p1,
    double *p2,
    float patm,
    float tf,

    float pb,
    float tb,

    float sg,
    float mu,
    float cpcv,
    float zf,

    short *flowmode,
    char *tablename
)
```

**Note...** \* Indicates a variable passed by reference (address), all other values passed by value. Calculated values will be returned in the associated “by reference” variables.

## Variable Definitions

**area = Orifice Area** - Specifies the orifice flow area for the relief valve. A pointer to a single precision floating point number greater than 0. Entered or calculated value. Only used by flow equation 12,13. Pass 0.0 if not used.

**calc\_var = Calculation Variable** - Specifies which variable to calculate the value of. Short integer. Valid values are:

- 1 = Valve Factor (cg) or Orifice Area
- 2 = Volumetric Flow Rate
- 3 = Inlet Pressure
- 4 = Outlet Pressure

**cg = Valve Factor** - Specifies the gas valve factor for the relief valve. A pointer to a single precision floating point number greater than 0. Entered or calculated value.

**cpcv = Specific Heat Ratio** - Specifies the ratio of the average specific heat at constant pressure to the average specific heat at constant volume of the gas. Single precision floating point number greater than 0 expressed in Decimal.

**eq\_code = Relief Valve Equation Code** - Specifies which relief valve flow equation to use. Short integer. Valid values are:

- 0 = Fisher Controls
- 1 = Rockwell
- 2 = Equimeter
- 3 = American AFV
- 4 = American
- 5 = ISA S75.01
- 6 = Grove 900
- 7 = Mooney Controls
- 8 = Groove 80
- 9 = Table Interpolation
- 10 = Pietro Fiorentini
- 12 = ASME BPV
- 13 = API 520
- 14 = Anderson Greenwood 90

**eq2\_code = Secondary Regulator Equation Code** - When eq\_code=10, specifies which regulator flow equation to use if the entered values are outside the limit of the table. Short integer. Valid values are the same as eq\_code.



**flowmode = Flow Mode** - Specifies the type of flow occurring in the regulator. A pointer to a short integer number. Calculated value. Possible return values are:

- 1 = Sub-critical/Sub-sonic flow
- 2 = Critical/Sonic flow
- 3 = Zero flow

**min\_dp = Minimum Differential Pressure** - Specifies the minimum differential pressure required to fully open the valve. Single precision floating point number greater than 0 expressed in Psia.

**mu = Viscosity** - Specifies the average absolute or dynamic viscosity of the gas. Single precision floating point number greater than 0 expressed in Lbm/Ft-Sec.

**p1 = Inlet Pressure** - Specifies the upstream (inlet) pressure of the relief valve. A pointer to a double precision floating point number greater than one negative atmosphere expressed in Psia. Entered or calculated value.

**p2 = Outlet Pressure** - Specifies the downstream (outlet) pressure of the relief valve. A pointer to a double precision floating point number greater than one negative atmosphere expressed in Psia. Entered or calculated value.

**parm\_1 = Parameter 1** - Specifies the value of a particular parameter associated with the specified sizing equation. The definition of the parameter varies according to the specified equation. See the following *Notes & Considerations* section for information about the parameter definitions.

**parm\_2 = Parameter 2** - Specifies the value of a particular parameter associated with the specified sizing equation. The definition of the parameter varies according to the specified equation. See the following *Notes & Considerations* section for information about the parameter definitions.

**parm\_3 = Parameter 3** - Specifies the value of a particular parameter associated with the specified sizing equation. The definition of the parameter varies according to the specified equation. See the following *Notes & Considerations* section for information about the parameter definitions.

**parm\_4 = Parameter 4** - Specifies the value of a particular parameter associated with the specified sizing equation. The definition of the parameter varies according to the specified equation. See the following *Notes & Considerations* section for information about the parameter definitions.

**patm = Atmospheric Pressure** - Specifies the calculated atmospheric pressure value at the elevation of the regulator. A single precision floating point number expressed in Psia. Only used with equation code 9. Pass 0.0 if not used.

**pb = Base Pressure** - Specifies the pressure basis for the volumetric flow rate (basis for a standard cubic foot). Single precision floating point number greater than 0 expressed in Psia.



**qf = Flow Rate** - Specifies the volumetric flow rate through the relief valve. A pointer to a double precision floating point number expressed in Cfh at the specified base pressure and temperature. Entered or calculated value.

**sg = Specific Gravity** - Specifies the average specific gravity of the gas (ideal gas value compared to dry air). Single precision floating point number greater than 0 expressed in Decimal.

**tf = Inlet Temperature** - Specifies the flowing gas temperature at the inlet to the relief valve. Single precision floating point number greater than 0 expressed in Rankine.

**tablename = Capacity Table Name** - Specifies the name of the capacity table for the associated relief valve. A pointer to a character string. Only used when the eq\_code value is 9. Pass an empty character string when not used.

**tb = Base Temperature** - Specifies the temperature basis for the volumetric flow rate (basis for a standard cubic foot). Single precision floating point number greater than 0 expressed in Rankine.

**zf = Compressibility** - Specifies the compressibility at the specified inlet conditions. A single precision floating point number, greater than 0, expressed in decimal. Pass 1.0 if unknown or unused. Not used by all equations.

### Return Values

---

The function returns a short integer number indicating the calculation status. Possible return values are as follows:

- 1 = Successful calculation.
- 0 = Not successful - unknown error.
  
- 1 = Invalid Valve Factor (cg).
- 2 = Invalid Flow Rate value.
- 3 = Invalid Inlet Pressure value.
- 4 = Invalid Pressure value - the inlet pressure must be greater than the outlet pressure.
- 5 = Invalid Inlet Temperature value.
- 6 = Invalid Specific Gravity value.
- 7 = Invalid Specific Heat Ratio value.
- 8 = Invalid Compressibility value.
- 9 = Minimum Differential Pressure value not met.
  
- 97 = Unsupported calculation variable.
- 98 = Unsupported equation code.
- 99 = Module not enabled or licensed.

---

## Licensing

---

Calculation Module: Pipe Flow

Additional Required Modules: None

---

## Notes & Considerations

---

- All values must be passed in listed dimensional units for the associated variable. Calculated values are returned in the listed dimensional units. The *CalcFact* or *CalcT* functions can be used to obtain the required conversion factor for non-standard dimensional units.
- With the exception of the base pressure value, where passed pressure values are expressed in absolute pressure (Psia) dimensional units, the atmospheric pressure value must be added to the gauge pressure value (Psig) before passing the variable to the calculation function. Conversely, where a returned (calculated) value is expressed in absolute pressure units, the atmospheric pressure must be subtracted from the returned value to obtain the gauge pressure value. The atmospheric pressure value may be calculated using the *CalcPatm* function.
- The definition for the parameter values varies according to the equation used. See the GASCalc Regulator Calculation Reference for a description of the various definitions.
- The general process for using this function is as follows:
  - Collect the required variable values from User-entered values.
  - Convert the entered variable values to the appropriate dimensional units.
  - Call the calculation function.
  - If no error code is returned, convert the returned values to the appropriate dimensional units for display to the User. If an error code is returned, display an appropriate message to the User.
  - Store, print, or use the input values and results as appropriate.
- When using equation code 9, the specified capacity table must be in GASCalc 5.0 format, specifically in table version GC.5.1 format. The GASCalc software can be used to create the required table.
- The *SetLicense* function must be called at least once before attempting to execute this calculation routine.
- See the appropriate GASCalc Calculation Reference for documentation and references for the calculations performed by this function.

## CalcReynolds - Reynolds Number Calculation

---

The CalcReynolds routine calculates the Reynolds Number value associated with a specified set of conditions.

### Function Syntax

---

```
CalcReynolds  
(  
    float d,  
    double qf,  
  
    float sg,  
    float mu,  
  
    float pb,  
    float tb  
)
```

### Variable Definitions

---

**d = Inside Diameter** - Specifies the equivalent inside diameter of the pipe or orifice that the gas is flowing through. A single precision floating point number greater than 0 expressed in Inches.

**mu = Viscosity** - Specifies the average absolute or dynamic viscosity of the gas. Single precision floating point number greater than 0 expressed in Lbm/Ft-Sec.

**pb = Base Pressure** - Specifies the pressure basis for the volumetric flow rate (basis for a standard cubic foot). Single precision floating point number greater than 0 expressed in Psia.

**qf = Flow Rate** - Specifies the volumetric flow rate through the pipe or orifice associated with the Inside Diameter value. A pointer to a double precision floating point number expressed in Cfh at the specified base pressure and temperature. Entered or calculated value.

**sg = Specific Gravity** - Specifies the average specific gravity of the gas (ideal gas value compared to dry air). Single precision floating point number greater than 0 expressed in Decimal.

**tb = Base Temperature** - Specifies the temperature basis for the volumetric flow rate (basis for a standard cubic foot). Single precision floating point number greater than 0 expressed in Rankine.



## Return Values

---

The function returns a short integer number indicating the calculation status. Possible return values are as follows:

- 1 = Invalid Viscosity value.
- 2 = Invalid Diameter value.
- 3 = Invalid Base Temperature value.
- 4 = Invalid Specific Gravity value.
- 5 = Invalid Base Pressure value.
- 6 = Invalid Flow Rate value.
  
- 99 = Module not enabled or licensed.
  
- X.X = The calculated Reynolds Number value.

## Licensing

---

Calculation Module: Pipe Flow  
Additional Required Modules: None

## Notes & Considerations

---

- All values must be passed in listed dimensional units for the associated variable. Calculated values are returned in the listed dimensional units. The *CalcFact* or *CalcT* functions can be used to obtain the required conversion factor for non-standard dimensional units.
- The general process for using this function is as follows:
  - Collect the required variable values from User-entered values.
  - Convert the entered variable values to the appropriate dimensional units.
  - Call the calculation function.
  - If no error code is returned, convert the returned values to the appropriate dimensional units for display to the User. If an error code is returned, display an appropriate message to the User.
  - Store, print, or use the input values and results as appropriate.
- The routine can be used to calculate the Reynolds Number for flow through any circular area where the inside diameter is known.
- The *SetLicense* function must be called at least once before attempting to execute this calculation routine.



- See the appropriate GASCalc Calculation Reference for documentation and references for the calculations performed by this function.

## CalcSonic - Sonic Velocity Calculation

The CalcSonic routine calculates the sonic velocity (speed of sound) for a specified gas composition and at specified conditions.

### Function Syntax

```
CalcSonic
(
    short calc_method,
    float pf,
    float tf,
    float pb,
    float tb,
    float patm,
    short z_method,
    float *pp,
    long interval,
    short *detail_cnt,
    char *detail
)
```

**Note...**

\* Indicates a variable passed by reference (address), all other values passed by value. Calculated values will be returned in the associated “by reference” variables.

### Variable Definitions

**calc\_method = Calculation Method** - Specifies the method for which to calculate the indicated variable. Short integer. Valid values are:

- 0 = GPSA (Traditional Thermodynamic) Method
- 1 = AGA 10 2003 Method
- 2 = AGA 8 2017 Detailed Method
- 3 = AGA 8 2017, GERG-2008 Method

**detail = Calculation Details** - Lists various intermediate and final calculation results. A pointer to a character array (string) at least 1280 characters in length. The format of the list is as follows:

```
Parameter 1 Name or Description
Parameter 1 Value
Value 1 Dimensional Units (if applicable)
.
.
.
Parameter detail_cnt Name or Description
Parameter detail_cnt Value
Value detail_cnt Dimensional Units (if applicable)
```

Each item in the list ends with a “new line” character.

**detail\_cnt** = Number of parameter entries in the Calculation Detail list. A pointer to a short integer.

**interval = Integration Interval** - Specifies the number of steps to include in the integration interval of the solution routine. Long integer value greater than 0. Not required when using calculation methods (calc\_method) 2 or 3. See the following *Notes & Considerations* section for additional information.

**patm = Atmospheric Pressure** - Specifies the calculated atmospheric pressure value associated with pf. A single precision floating point number greater than or equal to 0 expressed in Psia. Only required for certain compressibility calculation methods, pass 0 when not required.

**pb = Base Pressure** - Specifies the pressure basis for gas property values. Single precision floating point number greater than 0 expressed in Psia. Not required when using calculation methods (calc\_method) 2 or 3.

**pf = Pressure** - Specifies the flowing pressure value. Single precision floating point number greater than 0 expressed in Psia.

**pp = Gas Components** - Specifies the mole fraction of the various components in the gas mixture. A pointer to a single precision floating point number array. See the [CalcZF](#) function description for array requirements and definition. Or...

If using calculation method (calc\_method) 0:

pp[0] = Ratio of Specific Heats, dimensionless

pp[1] = Ideal Specific Gravity, dimensionless

**tb = Base Temperature** - Specifies the temperature basis for gas property values. Single precision floating point number greater than 0 expressed in Rankine. Not required when using calculation methods (calc\_method) 2 or 3.



**tf = Temperature** - Specifies the average flowing temperature of the gas. Single precision floating point number greater than 0 expressed in Rankine.

**z\_method = Compressibility Calculation Method** - Specifies the method for which to calculate the compressibility factor. Short integer.

If using calculation method (calc\_method) 0 or 1:

Valid values are:

0 = None

11 = Entered compressibility values - see the following *Notes & Considerations* section for required values.

*Note* - See the [CalcZF](#) function description for other acceptable values.

If using calculation methods (calc\_method) 2 or 3:

z\_method will be ignored. The compressibility method associated with the particular sonic velocity calculation (*i.e.*, 2 or 3) will be automatically calculated and used.

## Return Values

---

Upon success, the function returns a floating point number indicating the sonic velocity expressed in feet/sec. Or...

0 = Not successful - unknown error.

-1 = Invalid Calculation Method.

-99 = Module not enabled or licensed.

-100 = Error calculating Compressibility Factor or invalid Compressibility Calculation Method.

**Note...**

\* If the compressibility is calculated, other error codes may be returned if the compressibility calculation fails. See the [CalcZF](#) function description for additional possible return values.

## Licensing

---

Calculation Module: Sonic

Additional Required Modules: Compressibility\*

\*Only required if the Compressibility Calculation Method to be used is neither Method 0 nor 11.



## Notes & Considerations

- All passed values must be expressed in the listed dimensional units for the associated variable. Calculated values are returned in the listed dimensional units. The *CalcFact* or *CalcT* functions can be used to obtain the required conversion factor for non-standard dimensional units.

- With the exception of the base pressure value, where passed pressure values are expressed in absolute pressure (Psia) dimensional units, the atmospheric pressure value must be added to the gauge pressure value (Psig) before passing the variable to the calculation function. Conversely, where a returned (calculated) value is expressed in absolute pressure units, the atmospheric pressure must be subtracted from the returned value to obtain the gauge pressure value. The atmospheric pressure value may be calculated using the *CalcPatm* function.

- If the compressibility calculation method (*z\_method*) is set to 11, the various compressibility factors must be passed to the calculation function. The values are passed using the Gas Components array (*pp*) as follows:

If using calculation method (*calc\_method*) 0 or 1:

*pp*[0] = gas compressibility factor at flowing conditions (*pf* and *tf*).

- The general process for using this function is as follows:

- Collect the required variable values from User-entered values.
- Convert the entered variable values to the appropriate dimensional units.
- Call the calculation function.
- If no error code is returned, use or display the returned value to the User. If an error code is returned, display an appropriate message to the User.
- Store, print, or use the input values and results as appropriate.

- The *SetLicense* function must be called at least once before attempting to execute this calculation routine.

- The AGA 7 2003 calculation method recommends the use of the AGA 8 1992 Detailed compressibility calculation method. The routine allows other methods to be used at the User's discretion; however, they may yield questionable results.

- Calculation methods (*calc\_method*) 0 and 1 use an integration method to calculate the specific heat at a constant volume value. The interval parameter indicates the precision of this calculation. The smaller the number, the less precise the resulting calculated sonic velocity value. A value of 10000 is recommended for a sonic velocity precision of 1/100 of a meter/sec. Calculation methods (*calc\_method*) 2 and 3 do not use an integration method; hence, pass 0 for the interval when using either of these two methods.

- See the appropriate GASCalc Calculation Reference for documentation and references for the calculations performed by this function.



## CalcTherm - Thermodynamic Properties Calculation

The CalcTherm routine calculates individual thermodynamic property values for a specified gas composition at a specified temperature and pressure.

### Function Syntax

```
CalcTherm
(
    short calc_method,
    short calc_item,
    double pf,
    double tf,
    short z_method,
    double *pp,
    short *detail_cnt,
    char detail[1280],
    short specific_ok,
    double zf
)
```

**Note...**

\* Indicates a variable passed by reference (address), all other values passed by value. Calculated values will be returned in the associated “by reference” variables.

### Variable Definitions

**calc\_item = Calculated Item** - Specifies which individual thermodynamic property value to calculate and display. Only one calc\_item may be used at a time. Short integer. Valid values are:

1 = Enthalpy  $h$  - [J / mol] if specific\_ok = 0 and [kJ / kg] if specific\_ok = 1.

2 = Entropy  $S$  - [J / (mol\*K)] if specific\_ok = 0 and [kJ / (kg\*K)] if specific\_ok = 1.

3 = Isochoric Heat Capacity  $C_v$  - [J / (mol\*K)] if specific\_ok = 0 and [kJ / (kg\*K)] if specific\_ok = 1.

4 = Isobaric Heat Capacity  $C_p$  - [J / (mol\*K)] if specific\_ok = 0 and [kJ / (kg\*K)] if specific\_ok = 1.



5 = Joule-Thomson Coefficient  $\mu$  - [K / MPa]

6 = Molar Density  $d$  - [mol / dm<sup>3</sup>]

7 = Molar Mass (Molecular Weight)  $M$  - [g / mol]

8 = Mass Density (Mass Per Unit Volume)  $\rho$  - [Lbm / cf]

**calc\_method** = **Calculation Method** - Specifies the method for which to calculate the indicated variable. Short integer. Valid values are:

1 = AGA 8 2017 Detailed Method

2 = AGA 8 2017, GERG-2008 Method

**pf** = **Pressure** - Specifies the flowing pressure value. Double precision floating point number greater than 0 expressed in Psia.

**pp** = **Gas Components** - Specifies the mole fraction of the various components in the gas mixture. A pointer to a double precision floating point number array. See the [CalcZF](#) function description for array requirements and definitions.

**specific\_ok** = Specifies which type of units to display for calc\_items 1, 2, 3, and 4, *i.e.*, Enthalpy, Entropy,  $C_v$ , &  $C_p$ , respectively. Short integer. Valid values are:

0 = Units for calc\_item 1-4 will be molar, *i.e.*, the returned value will *not* be divided by the molar mass (molecular weight) of the mixture before being displayed.

1 = Units for calc\_item 1-4 will *not* be molar, *i.e.*, the returned value will be divided by the molar mass (molecular weight) of the mixture before being displayed.

See the calc\_item 1-4 Variable Definitions above for more information regarding the specific units which depend on the value of specific\_ok.

The only relevant return values for specific\_ok are calc\_items 1-4. In other words, the only situation for which the value of specific\_ok is relevant is when the User is calculating calc\_items 1-4. Otherwise, specific\_ok is ignored.

**tf** = **Temperature** - Specifies the average flowing temperature of the gas. Double precision floating point number greater than 0 expressed in Rankine.

**zf** = **Compressibility Factor** - Specifies the manually entered gas compressibility factor at flowing conditions (pf and tf). Double precision floating point number greater than 0. Dimensionless. Only relevant when z\_method = 11. If z\_method is not 11, zf is ignored.



**z\_method = Compressibility Calculation Method** - Specifies the method for which to calculate or enter the compressibility factor. Short integer. Valid values are:

- 0 = None
- 11 = Entered compressibility values

*Note* - Unless **z\_method** = 11, **z\_method** will be ignored. If **z\_method** is not 11, the compressibility method associated with the specific thermodynamic property calculation (*i.e.*, 1 or 2) will be used.

*Note* - The only relevant return value for **z\_method** is rho ( $\rho$ ) (*i.e.*, **calc\_item** = 8). In other words, the only situation for which the value of **z\_method** matters is when the User is calculating rho.

## Return Values

Upon success, the function returns a double precision floating point number indicating the value of the desired calculated item (**calc\_item**) in the specified units. Or...

- 0 = Not successful - unknown error.
- 1 = Invalid Calculation Method.
- 2 = Invalid Calculation Item.
- 4 = Invalid pressure value (*i.e.*,  $pf \leq 0$ )
- 5 = Invalid temperature value (*i.e.*,  $tf \leq 0$ )
- 99 = Module not enabled or licensed.
- 100 = Error calculating Compressibility Factor or invalid Compressibility Calculation Method.
- 133 = Sum of mole fractions does not equal 1.
- 141 = Invalid Molar Density (*i.e.*,  $d \leq 0$ )

## Licensing

Calculation Module: Compressibility  
Additional Required Modules: None

## Notes & Considerations

- All passed values must be expressed in the listed dimensional units for the associated variable. Calculated values are returned in the listed dimensional units. The *CalcFact* or *CalcT* functions can be used to obtain the required conversion factor for non-standard dimensional units.



- With the exception of the base pressure value, where passed pressure values are expressed in absolute pressure (Psia) dimensional units, the atmospheric pressure value must be added to the gauge pressure value (Psig) before passing the variable to the calculation function. Conversely, where a returned (calculated) value is expressed in absolute pressure units, the atmospheric pressure must be subtracted from the returned value to obtain the gauge pressure value. The atmospheric pressure value may be calculated using the *CalcPatm* function.

- If the compressibility calculation method (z\_method) is set to 11:

- The compressibility factor must be passed to the calculation function. The value is passed using the *zf* parameter. That is,

*zf* = manually entered gas compressibility factor at flowing conditions (*pf* and *tf*).

- The general process for using this function is as follows:

- Collect the required variable values from the User-entered values.
  - Convert the entered variable values to the appropriate dimensional units.
  - Call the calculation function.
  - If no error code is returned, use or display the returned value to the User. If an error code is returned, display an appropriate message to the User.
  - Store, print, or use the input values and results as appropriate.

- The *SetLicense* function must be called at least once before attempting to execute this calculation routine.

## CalcT - Temperature Unit Conversion

The CalcT routine calculates a temperature value in certain specified dimensional units to a value in other specified dimensional units.

### Function Syntax

```
CalcT  
(  
    char *from_units,  
    char *to_units,  
    float t  
)
```

*Note...*

\* Indicates a variable passed by reference (address), all other values passed by value. Calculated values will be returned in the associated “by reference” variables.

### Variable Definitions

**from\_units = From Units** - Specifies dimensional units for the value of variable **t**. A pointer to a character (string) value. Valid values are:

“Celsius”, “C”  
“Fahrenheit”, “F”  
“Kelvin”, “K”  
“Rankine”, “R”

**t = Temperature Value** - The value to be converted. Single precision floating point number expressed in from\_units. Any valid temperature value.

**to\_units = To Units** - Specifies the dimensional units to convert the variable **t** value to. A pointer to a character (string) value. Valid values are:

“Celsius”, “C”  
“Fahrenheit”, “F”  
“Kelvin”, “K”  
“Rankine”, “R”



### Return Values

---

The function returns a single precision floating point number which represents the value of **t** expressed in to\_units dimensional units. Or returns -99 if the module is not enabled or licensed.

### Licensing

---

Calculation Module: Support  
Additional Required Modules: None

### Notes & Considerations

---

- Use the *CalcFact* routine to convert other value types.
- The *SetLicense* function must be called at least once before attempting to execute this calculation routine.

## CalcUSM - Ultrasonic Meter Calculation

The CalcUSM routine calculates various values associated with the gas flow through an ultrasonic-type flow measurement meter.

### Function Syntax

```
CalcUSM
(
    short calc_method,
    short calc_var,
    float din,
    float dout,
    float *pf,
    double *qb,
    float *vel,
    float tf,
    float pb,
    float tb,
    short z_method,
    float *pp,
    float tref,
    float pref,
    float alpha,
    double modulus,
    float poisson,
    float mfactor,
    short *detail_cnt,
    char *detail
)
```

#### Note...

\* Indicates a variable passed by reference (address), all other values passed by value. Calculated values will be returned in the associated “by reference” variables.

### Variable Definitions

**alpha = Meter Body Expansion Factor** - Specifies the coefficient of linear thermal expansion for the meter body material. Single precision floating point number expressed in In/In-F. Any valid expansion value.



**calc\_method** = **Calculation Method** - Specifies the method to use to calculate the indicated variable. Short integer. Valid values are:

2 = Daniels Meter Method

**calc\_var** = **Variable Calculation Flag** - Specifies which variable is to be calculated. Short integer. Valid values are:

2 = Volumetric Flow Rate

**detail** = **Calculation Details** - Lists various intermediate and final calculation results. A pointer to a character (string) value at least 1280 characters in length. The format of the list is as follows:

```
Parameter 1 Name or Description
Parameter 1 Value
Value 1 Dimensional Units (if applicable)
.
.
.
Parameter detail_cnt Name or Description
Parameter detail_cnt Value
Value detail_cnt Dimensional Units (if applicable)
```

Each item in the list ends with a “new line” character. (ASCII Character 10)

**detail\_cnt** = Number of parameter entries in the Calculation Detail list. A pointer to a short integer. Pass as -1 to suppress generation of the calculation details.

**din** = **Meter Inside Diameter** - Specifies the inside diameter of the meter body at the specified reference temperature tref and reference pressure pref. Single precision floating point number greater than 0 expressed in Inches.

**dout** = **Meter Outside Diameter** - Specifies the outside diameter of the meter body at the specified reference temperature tref and reference pressure pref. Single precision floating point number greater than 0 expressed in Inches.

**mfactor** = **Meter Factor** - Specifies the meter factor associated with the meter body material. Single precision floating point number greater than 0 - normally 1.0. Dimensionless.

**modulus** = **Young’s Modulus** - Specifies the Young’s Modulus (Modulus of Elasticity) of the meter body material. Double precision floating point number greater than 0 expressed in Psi.

**pb** = **Base Pressure** - Specifies the pressure basis for stated volumetric flow rates (basis for a standard cubic foot). Single precision floating point number greater than 0 expressed in Psia.



**pf = Pressure** - Specifies the static pressure value within the meter body. Single precision floating point number greater than 0 expressed in Psia.

**poisson = Poisson Ratio** - Specifies the Poisson Ratio of the meter body material. Single precision floating point number greater than 0. Dimensionless.

**pref = Reference Pressure** - Specifies the reference pressure value for the meter inside (din) and outside (dout) diameter values. Single precision floating point number greater than 0 expressed in Psia.

**pp = Gas Components** - Specifies the mole fraction of the various components in the gas mixture. A pointer to single precision floating point number array. See the [CalcZF](#) function description for array requirements and definition. Or...

If using the compressibility calculation method (z\_method) 11:

pp[0] = gas compressibility factor at flowing conditions (pf and tf).

pp[1] = gas compressibility factor at base conditions (pb and tb).

pp[2] = dry air only compressibility factor at base conditions (pb and tb).

pp[3] = gas compressibility factor at standard conditions (ps and ts).

If using the compressibility calculation method (z\_method) 21:

pp[0] = Supercompressibility correction factor (Fpv)

**qb = Flow Rate** - Specifies the volumetric flow rate through the orifice. A pointer to a double precision floating point number expressed in Cfh at the specified base pressure and temperature. This variable can be specified (passed) or calculated. If specified, the value must be greater than 0.

**tb = Base Temperature** - Specifies the temperature basis for stated volumetric flow rates (basis for a standard cubic foot). Single precision floating point number greater than 0 expressed in Rankine.

**tf = Temperature** - Specifies the average temperature of the gas. Single precision floating point number greater than 0 expressed in Rankine.

**trref = Reference Temperature** - Specifies the reference temperature for the meter inside (din) and outside (dout) diameter values. Single precision floating point number expressed in Rankine. Any valid temperature value.

**vel = Velocity** - Specifies the flow velocity through the meter. A pointer to a single precision floating point number expressed in Ft/sec. This variable can be specified (passed) or calculated. If specified, the value must be greater than 0.

**z\_method = Compressibility Calculation Method** - Specifies the method to use to calculate the compressibility factor. Short integer. Valid values are:

- 0 = None
- 11 = Entered compressibility values
- 21 = Entered Supercompressibility correction factor (Fpv)

Or other values associated with the various calculation methods supported by the *CalcZF* function. See the [CalcZF](#) function description for other acceptable values.

### Return Values

The function returns a short integer number indicating the calculation status. Possible return values are as follows:

- 1 = Successful calculation.
- 0 = Not successful - unknown error.
  
- 1 = Unsupported compressibility calculation method.
- 3 = Beta Ratio too small.
- 4 = Beta Ratio too large.
- 5 = Reynolds Number too small.
- 6 = Orifice Size too small.
- 7 = Error calculating flow rate solution.
- 8 = Unable to calculate unknown value - zero flow value.
- 9 = Error calculating orifice size solution.
- 10 = Error calculating differential pressure solution.
- 98 = Unsupported calculation method.
- 99 = Module not enabled or licensed.

**Note...**

\* If the compressibility is calculated, other error codes may be returned if the compressibility calculation fails. See the [CalcZF](#) function description for additional possible return values.

## Licensing

---

Calculation Module: Meter

Additional Required Modules: Compressibility\*

\*Only required if the Compressibility Calculation Method to be used is other than Method 0, 11, or 21.

## Notes & Considerations

---

- All passed values must be expressed in the listed dimensional units for the associated variable. Calculated values are returned in the listed dimensional units. The *CalcFact* or *CalcT* functions can be used to obtain the required conversion factor for non-standard dimensional units.
- With the exception of the base pressure value, where passed pressure values are expressed in absolute pressure (Psia) dimensional units, the atmospheric pressure value must be added to the gauge pressure value (Psig) before passing the variable to the calculation function. Conversely, where a returned (calculated) value is expressed in absolute pressure units, the atmospheric pressure must be subtracted from the returned value to obtain the gauge pressure value. The atmospheric pressure value may be calculated using the *CalcPatm* function.
- The general process for using this function is as follows:
  - Collect the required variable values from User-entered values.
  - Convert the entered variable values to the appropriate dimensional units.
  - Call the calculation function.
  - If no error code is returned, use or display the returned value to the User. If an error code is returned, display an appropriate message to the User.
  - Store, print, or use the input values and results as appropriate.
- The *SetLicense* function must be called at least once before attempting to execute this calculation routine.
- See the appropriate GASCalc Calculation Reference for documentation and references for the calculations performed by this function.



## CalcVCone - V-Cone Meter Calculation

The CalcVCone routine calculates various values associated with gas flow through a v-cone-type flow measurement meter.

### Function Syntax

```
CalcVCone
(
    short calc_method,
    short calc_var,
    float dr,
    float *orf,
    float *dp,
    double *qf,
    float pf,
    float tf,
    float pb,
    float tb,
    float patm,
    float sg,
    short sg_real,
    float mu,
    float cpcv,
    short z_method,
    float *pp,
    float tr_orf,
    float tr_tube,
    float x_orf,
    float x_tube,
    float cd,
    short *detail_cnt,
    char *detail
)
```

**Note...**

\* Indicates a variable passed by reference (address), all other values passed by value. Calculated values will be returned in the associated “by reference” variables.

## Variable Definitions

**calc\_method** = **Calculation Method** - Specifies the method to use to calculate the indicated variable. Short integer. Valid values are:

- 1 = McCrometer With Adiabatic Expansion Method
- 2 = McCrometer With V-Cone Expansion Method
- 3 = McCrometer With Wafer-Cone Expansion Method

**calc\_var** = **Variable Calculation Flag** - Specifies which variable is to be calculated. Short integer. Valid values are:

- 3 = Volumetric Flow Rate

**cd** = **Discharge Coefficient** - Specifies the coefficient of discharge for the meter. Single precision floating point number greater than 0 expressed in Decimal.

**cpcv** = **Specific Heat Ratio** - Specifies the ratio of the average specific heat at constant pressure to the average specific heat at constant volume of the gas. Single precision floating point number greater than 0 expressed in Decimal.

**detail** = **Calculation Details** - Lists various intermediate and final calculation results. A pointer to a character (string) value at least 1280 characters in length. The format of the list is as follows:

```
Parameter 1 Name or Description
Parameter 1 Value
Value 1 Dimensional Units (if applicable)
.
.
.
Parameter detail_cnt Name or Description
Parameter detail_cnt Value
Value detail_cnt Dimensional Units (if applicable)
```

Each item in the list ends with a “new line” character.

**detail\_cnt** = Number of parameter entries in the Calculation Detail list. A pointer to a short integer.

**dp** = **Differential Pressure** - Specifies the pressure differential across the meter cone. A pointer to a single precision floating point number expressed in Psi. This variable can be specified (passed) or calculated. If specified, the value must be greater than 0.

**dr** = **Meter Tube Diameter** - Specifies the inside diameter of the meter tube (pipe) at the specified reference temperature tr\_tube. Single precision floating point number greater than 0 expressed in Inches.



**mu = Viscosity** - Specifies the absolute or dynamic viscosity of the gas. Single precision floating point number greater than 0 expressed in Lbm/Ft-Sec.

**orf = Orifice Diameter** - Specifies the diameter of the orifice at the specified reference temperature tr\_orf. A pointer to a single precision floating point number expressed in Inches. This variable can be specified (passed) or calculated. If specified the value must be greater than 0.

**patm = Atmospheric Pressure** - Specifies the calculated atmospheric pressure value associated with pf. A single precision floating point number greater than or equal to 0 expressed in Psia. Only required for certain compressibility calculation methods, pass 0 when not required.

**pb = Base Pressure** - Specifies the pressure basis for stated volumetric flow rates (basis for a standard cubic foot). Single precision floating point number greater than 0 expressed in Psia.

**pf = Pressure** - Specifies the static pressure value. Single precision floating point number greater than 0 expressed in Psia.

**pp = Gas Components** - Specifies the mole fraction of the various components in the gas mixture. A pointer to single precision floating point number array. See the [CalcZF](#) function description for array requirements and definition.

**qf = Flow Rate** - Specifies the volumetric flow rate through the orifice. A pointer to a double precision floating point number expressed in Cfh at the specified base pressure and temperature. This variable can be specified (passed) or calculated. If specified, the value must be greater than 0.

**sg = Specific Gravity** - Specifies the average specific gravity of the gas (ideal or real value compared to dry air). Single precision floating point number greater than 0 expressed in Decimal. Only required for certain compressibility calculation methods, pass 0 when not required. Equals actual gas density expressed in Lbm/ft<sup>3</sup> when sg\_real=2.

**sg\_real = Specific Gravity Type** - Specifies the specific gravity type. Short integer. Valid values are:

- 0 = Ideal gas value.
- 1 = Real gas value.
- 2 = Actual gas density value.

**tb = Base Temperature** - Specifies the temperature basis for stated volumetric flow rates (basis for a standard cubic foot). Single precision floating point number greater than 0 expressed in Rankine.

**tf = Temperature** - Specifies the average temperature of the gas. Single precision floating point number greater than 0 expressed in Rankine.

**tr\_orf = Orifice Reference Temperature** - Specifies the reference temperature for the orfr variable value. Single precision floating point number expressed in Rankine. Any valid temperature value.



**tr\_tube = Meter Tube Reference Temperature** - Specifies the reference temperature for the dr variable value. Single precision floating point number expressed in Rankine. Any valid temperature value.

**x\_orf=Orifice Expansion Factor** - Specifies the coefficient of linear thermal expansion for the orifice plate material. Single precision floating point number expressed in In/In-F. Any valid expansion value.

**x\_tube = Meter Tube Expansion Factor** - Specifies the coefficient of linear thermal expansion for the meter tube material. Single precision floating point number expressed in In/In-F. Any valid expansion value.

**z\_method = Compressibility Calculation Method Flag** - Specifies the method to use to calculate the compressibility factor. Ignored if sg\_real=2. Short integer. Valid values are:

0 = None

11 = Entered compressibility values - see the following *Notes & Considerations* section for required values.

*Note* - See the [CalcZF](#) function description for other acceptable values.

## Return Values

---

The function returns a short integer number indicating the calculation status. Possible return values are as follows:

1 = Successful calculation.

0 = Not successful - unknown error.

-99 = Module not enabled or licensed.

*Note* - If the compressibility is calculated, other error codes may be returned if the compressibility calculation fails. See the [CalcZF](#) function description for additional possible return values.

## Licensing

---

Calculation Module: Meter

Additional Required Modules: Compressibility\*

\*Only required if the compressibility calculation method to be used is other than Method 0 or 11.



## Notes & Considerations

- All passed values must be expressed in the listed dimensional units for the associated variable. Calculated values are returned to the listed dimensional units. The *CalcFact* or *CalcT* functions can be used to obtain the required conversion factor for non-standard dimensional units.

- With the exception of the base pressure value, where passed pressure values are expressed in absolute pressure (Psia) dimensional units, the atmospheric pressure value must be added to the gauge pressure value (Psig) before passing the variable to the calculation function. Conversely, where a returned (calculated) value is expressed in absolute pressure units, the atmospheric pressure must be subtracted from the returned value to obtain the gauge pressure value. The atmospheric pressure value may be calculated using the *CalcPatm* function.

- If the compressibility calculation method (*z\_method* variable) is set to 11, the various compressibility factors must be passed to the calculation function. The values are passed using the Gas Components array (*pp* variable) as follows:

*pp*[0] = gas compressibility factor at flowing conditions (*pf* and *tf*).

*pp*[1] = gas compressibility factor at base conditions (*pb* and *tb*).

*pp*[2] = dry air only compressibility factor at base conditions (*pb* and *tb*).

- The general process for using this function is as follows:

- Collect the required variable values from User-entered values.
- Convert the entered variable values to the appropriate dimensional units.
- Call the calculation function.
- If no error code is returned, use or display the returned value to the User. If an error code is returned, display an appropriate message to the User.
- Store, print, or use the input values and results as appropriate.

- The *SetLicense* function must be called at least once before attempting to execute this calculation routine.

- See the appropriate GASCalc Calculation Reference for documentation and references for the calculations performed by this function.

## CalcZF - Compressibility Calculation

The CalcZF routine calculates the compressibility factor for a specified set of conditions and gas properties.

### Function Syntax

```
CalcZF
(
    short calc_method,
    float pf,
    float tf,
    float pb,
    float tb,
    float patm,
    float sg,
    float *pp,
    float *roe,
    short ck_ok
)
```

**Note...**

\* Indicates a variable passed by reference (address), all other values passed by value. Calculated values will be returned in the associated “by reference” variables.

### Variable Definitions

**calc\_method = Calculation Method** - Specifies the method for which to calculate the compressibility factor. Short integer. Valid values are:

- 1 = AGA 8 1992 Detailed Method
- 2 = AGA 8 1992 Gross Method 1
- 3 = AGA 8 1992 Gross Method 2
- 4 = AGA NX 19 Standard Method
- 11 = Entered compressibility values - used by certain calculation routines.
- 12 = GPA 2172-09 Simplified Method
- 13 = GPA 2172-09 Rigorous Method
- 14 = AGA 8 2017, GERG-2008 Method
- 15 = AGA 8 1985 Primary Method
- 16 = AGA 8 1985 Alternate Method 1
- 17 = AGA 8 1985 Alternate Method 2



18 = AGA 8 1985 Alternate Method 3  
19 = AGA 8 1985 Alternate Method 4  
20 = AGA 8 1985 Alternate Method 5  
21 = Entered Supercompressibility Correction Factor - used by certain calculation routines.  
24 = AGA 8 2017 Detailed Method  
25 = AGA 8 2017 Gross Method 0  
26 = AGA 8 2017 Gross Method 1  
27 = AGA 8 2017 Gross Method 2  
28 = AGA 8 2017 Gross Method 3  
29 = International Standard ISO 6976 2016

**ck\_ok = Bounds Check Flag** - Indicates whether the various value bounds checks should be performed during the calculation. Valid values are:

0 = Do Not Perform Check  
1 = Perform Normal Check (Strictest Check)  
2 = Perform Expanded Check

**patm = Atmospheric Pressure** - Specifies the calculated atmospheric pressure value associated with pf. A single precision floating point number greater than 0 expressed in Psia. Only required for certain compressibility calculation methods, pass 0 when not required. Required for Methods 4, 12, and 13.

**pb = Base Pressure** - Specifies the pressure basis for stated volumetric flow rates (basis for a standard cubic foot). Single precision floating point number greater than 0 expressed in Psia. Required for Methods 2, 26, and 27.

**pf = Pressure** - Specifies the pressure value for which to calculate the compressibility factor. Single precision floating point number greater than 0 expressed in Psia.

**pp = Gas Components** - Specifies the mole fraction of the various components in the gas mixture. A pointer to single precision floating point number array. The array must be dimensioned to accommodate at least 24 components. Individual component values must be between 0.0 and 1.0. The component order/assignment is listed in the following tables depending on the selected calculation method.

Calculation Method 1, AGA 8 1992 - Detailed  
Calculation Method 12, GPA 2172-09 Simplified  
Calculation Method 14, AGA 8 2017 - GERG-2008  
Calculation Method 15, AGA 8 1985 - Primary  
Calculation Method 24, AGA 8 2017 - Detailed  
Calculation Method 29 - International Standard ISO 6976 2016

| Array Index | Component        | Molecular Formula                               |
|-------------|------------------|-------------------------------------------------|
| 0           | Methane          | CH <sub>4</sub>                                 |
| 1           | Ethylene         | C <sub>2</sub> H <sub>4</sub>                   |
| 2           | Ethane           | C <sub>2</sub> H <sub>6</sub>                   |
| 3           | Propane          | C <sub>3</sub> H <sub>8</sub>                   |
| 4           | Iso-Butane       | iC <sub>4</sub> H <sub>10</sub>                 |
| 5           | Normal-Butane    | nC <sub>4</sub> H <sub>10</sub>                 |
| 6           | Iso-Pentane      | iC <sub>5</sub> H <sub>12</sub>                 |
| 7           | Normal-Pentane   | nC <sub>5</sub> H <sub>12</sub>                 |
| 8           | Hexane           | C <sub>6</sub> H <sub>14</sub>                  |
| 9           | Heptane          | C <sub>7</sub> H <sub>16</sub>                  |
| 10          | Octane           | C <sub>8</sub> H <sub>18</sub>                  |
| 11          | Nonane           | C <sub>9</sub> H <sub>20</sub>                  |
| 12          | Decane           | C <sub>10</sub> H <sub>22</sub>                 |
| 13          | Air              | N <sub>2</sub> +O <sub>2</sub><br>(Approximate) |
| 14          | Water            | H <sub>2</sub> O                                |
| 15          | Argon            | Ar                                              |
| 16          | Carbon Monoxide  | CO                                              |
| 17          | Carbon Dioxide   | CO <sub>2</sub>                                 |
| 18          | Hydrogen Sulfide | H <sub>2</sub> S                                |
| 19          | Helium           | He                                              |
| 20          | Hydrogen         | H <sub>2</sub>                                  |
| 21          | Nitrogen         | N <sub>2</sub>                                  |
| 22          | Oxygen           | O <sub>2</sub>                                  |
| 23          | Other            |                                                 |

Calculation Method 2, AGA 8 1992 - Gross Method 1

| Array Index | Item                   | Dimensional Units |
|-------------|------------------------|-------------------|
| 0           | Heating Value          | Btu/cf            |
| 1           | Specific Gravity       |                   |
| Array Index | Component              | Molecular Formula |
| 2           | Carbon Dioxide         | CO <sub>2</sub>   |
| 5-23        | Not used, must equal 0 |                   |

Calculation Method 3, AGA 8 1992 - Gross Method 2

Calculation Method 4, AGA NX19 Standard

| Array Index | Item                    | Dimensional Units |
|-------------|-------------------------|-------------------|
| 1           | Specific Gravity (Real) |                   |
| Array Index | Component               | Molecular Formula |
| 2           | Carbon Dioxide          | CO <sub>2</sub>   |
| 3           | Nitrogen                | N <sub>2</sub>    |
| All others  | Not used, must equal 0  |                   |

Calculation Method 11, Entered Compressibility Values

| Array Index | Item                                                                                                                   |
|-------------|------------------------------------------------------------------------------------------------------------------------|
| 0           | Compressibility of gas at flowing conditions (Zf)                                                                      |
| 1           | Compressibility of gas at base conditions (Zb)                                                                         |
| 2           | Compressibility of dry air at base or standard conditions (ZbAir, ZsAir) depending on the calculation being used with. |
| 3           | Compressibility of gas at standard conditions (Zs)                                                                     |
| All others  | Not used, must equal 0                                                                                                 |

## Calculation Method 16, AGA 8 1985 - Alternate Method 1

| Array Index | Item                                                           | Dimensional Units |
|-------------|----------------------------------------------------------------|-------------------|
| 14          | Specific Gravity                                               |                   |
| 15          | Specific Gravity Type.<br>0 = Ideal (un-supported)<br>1 = Real |                   |
| Array Index | Component                                                      | Molecular Formula |
| 17          | Carbon Dioxide                                                 | CO2               |
| 21          | Nitrogen                                                       | N2                |
| All others  | Not used, must equal 0                                         |                   |

## Calculation Method 17, AGA 8 1985 - Alternate Method 2

| Array Index | Item                                                           | Dimensional Units |
|-------------|----------------------------------------------------------------|-------------------|
| 14          | Specific Gravity                                               |                   |
| 15          | Specific Gravity Type.<br>0 = Ideal (un-supported)<br>1 = Real |                   |
| 16          | Heating Value                                                  | Btu/cf            |
| Array Index | Component                                                      | Molecular Formula |
| 17          | Carbon Dioxide                                                 | CO2               |
| 21          | Nitrogen                                                       | N2                |
| All others  | Not used, must equal 0                                         |                   |

## Calculation Method 18, AGA 8 1985 - Alternate Method 3

| Array Index | Item                                                           | Dimensional Units |
|-------------|----------------------------------------------------------------|-------------------|
| 14          | Specific Gravity                                               |                   |
| 15          | Specific Gravity Type.<br>0 = Ideal (un-supported)<br>1 = Real |                   |
| 16          | Heating Value                                                  | Btu/cf            |
| Array Index | Component                                                      | Molecular Formula |
| 17          | Carbon Dioxide                                                 | CO2               |
| All others  | Not used, must equal 0                                         |                   |

## Calculation Method 19, AGA 8 1985 - Alternate Method 4

| Array Index | Item                   | Dimensional Units |
|-------------|------------------------|-------------------|
| 16          | Heating Value          | Btu/cf            |
| Array Index | Component              | Molecular Formula |
| 17          | Carbon Dioxide         | CO2               |
| 21          | Nitrogen               | N2                |
| All others  | Not used, must equal 0 |                   |

## Calculation Method 20, AGA 8 1985 - Alternate Method 5

| Array Index | Item                                                           | Dimensional Units |
|-------------|----------------------------------------------------------------|-------------------|
| 14          | Specific Gravity                                               |                   |
| 15          | Specific Gravity Type.<br>0 = Ideal (un-supported)<br>1 = Real |                   |

| Array Index | Component              | Molecular Formula |
|-------------|------------------------|-------------------|
| 0           | Methane                | CH <sub>4</sub>   |
| 17          | Carbon Dioxide         | CO <sub>2</sub>   |
| 21          | Nitrogen               | N <sub>2</sub>    |
| All others  | Not used, must equal 0 |                   |

Calculation Method 21, Entered Supercompressibility Correction Factor Value

| Array Index | Item                                         |
|-------------|----------------------------------------------|
| 0           | Supercompressibility Correction Factor (Fpv) |
| All others  | Not used, must equal 0                       |

Calculation Method 25, AGA 8 2017 - Gross Method 0

Calculation Method 13, GPA 2172-09 Rigorous

| Array Index | Component                                                  | Molecular Formula                                                                                                                                                   |
|-------------|------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0           | Methane                                                    | CH <sub>4</sub>                                                                                                                                                     |
| 1           | Ethylene                                                   | C <sub>2</sub> H <sub>4</sub>                                                                                                                                       |
| 2           | Ethane                                                     | C <sub>2</sub> H <sub>6</sub>                                                                                                                                       |
| 3           | Propane                                                    | C <sub>3</sub> H <sub>8</sub>                                                                                                                                       |
| 4           | Iso-Butane                                                 | iC <sub>4</sub> H <sub>10</sub>                                                                                                                                     |
| 5           | Normal-Butane                                              | nC <sub>4</sub> H <sub>10</sub>                                                                                                                                     |
| 6           | Iso-Pentane                                                | iC <sub>5</sub> H <sub>12</sub>                                                                                                                                     |
| 7           | Normal-Pentane                                             | nC <sub>5</sub> H <sub>12</sub>                                                                                                                                     |
| 8           | Hexanes Plus (Hexane + Heptane + Octane + Nonane + Decane) | C <sub>6</sub> H <sub>14</sub> + C <sub>7</sub> H <sub>16</sub> + C <sub>8</sub> H <sub>18</sub> + C <sub>9</sub> H <sub>20</sub> + C <sub>10</sub> H <sub>22</sub> |
| 9           | Not used, must equal 0                                     |                                                                                                                                                                     |

| Array Index | Component              | Molecular Formula      |
|-------------|------------------------|------------------------|
| 10          | Not used, must equal 0 |                        |
| 11          | Not used, must equal 0 |                        |
| 12          | Not used, must equal 0 |                        |
| 13          | Air                    | N2+O2<br>(Approximate) |
| 14          | Water                  | H2O                    |
| 15          | Argon                  | Ar                     |
| 16          | Carbon Monoxide        | CO                     |
| 17          | Carbon Dioxide         | CO2                    |
| 18          | Hydrogen Sulfide       | H2S                    |
| 19          | Helium                 | He                     |
| 20          | Hydrogen               | H2                     |
| 21          | Nitrogen               | N2                     |
| 22          | Oxygen                 | O2                     |
| 23          | Other                  |                        |

Calculation Method 26, AGA 8 2017 - Gross Method 1

| Array Index | Item                                                           | Dimensional Units |
|-------------|----------------------------------------------------------------|-------------------|
| 9           | Heating Value                                                  | Btu/cf            |
| 10          | Specific Gravity                                               |                   |
| Array Index | Component                                                      | Molecular Formula |
| 11          | Specific Gravity Type.<br>0 = Ideal (un-supported)<br>1 = Real |                   |
| 17          | Carbon Dioxide                                                 | CO2               |
| All others  | Not used, must equal 0                                         |                   |

## Calculation Method 27, AGA 8 2017 - Gross Method 2

| Array Index | Item                                                           | Dimensional Units |
|-------------|----------------------------------------------------------------|-------------------|
| 10          | Specific Gravity                                               |                   |
| 11          | Specific Gravity Type.<br>0 = Ideal (un-supported)<br>1 = Real |                   |
| Array Index | Component                                                      | Molecular Formula |
| 17          | Carbon Dioxide                                                 | CO2               |
| 21          | Nitrogen                                                       | N2                |
| All others  | Not used, must equal 0                                         |                   |

## Calculation Method 28, AGA 8 2017 - Gross Method 3

| Array Index | Item                                                                                                                      | Dimensional Units |
|-------------|---------------------------------------------------------------------------------------------------------------------------|-------------------|
| 10          | Specific Gravity<br>(The molar mass of the mixture is calculated using the specific gravity and the molar weight of air). |                   |
| 11          | Specific Gravity Type.<br>0 = Ideal (un-supported)<br>1 = Real                                                            |                   |
| Array Index | Component                                                                                                                 | Molecular Formula |
| 17          | Carbon Dioxide                                                                                                            | CO2               |
| 21          | Nitrogen                                                                                                                  | N2                |
| All others  | Not used, must equal 0                                                                                                    |                   |

**roe** = Density. Specifies the calculated gas density and specified conditions. A pointer to a single precision floating point number expressed in Lbm/Cf. Calculated value. Not calculated in Methods 4, 12, and 13.

**sg** = Specific Gravity. Specifies the average specific gravity of the gas (ideal value compared to dry air). Single precision floating point number greater than 0 expressed in Decimal. Only required for certain compressibility calculation methods, pass 0 when not required.

**tb** = Base Temperature. Specifies the temperature basis for stated volumetric flow rates (basis for a standard cubic foot). Single precision floating point number greater than 0 expressed in Rankine. Required for Methods 2, 26, and 27.

**tf** = Temperature. Specifies the gas temperature for which to calculate the compressibility factor. Single precision floating point number greater than 0 expressed in Rankine.

### Return Values

The function returns a double precision number indicating the calculated compressibility factor. Or...

- 1 = Unsupported calculation method.
- 99 = Module is not enabled or licensed.
- 100 = Temperature out-of-range.
- 101 = Pressure out-of-range.
- 102 = Excessive Hydrogen Sulfide content.
- 103 = Excessive Hydrogen content.
- 104 = Excessive Carbon Monoxide content.
- 105 = Excessive Hexane content.
- 106 = Excessive Helium content.
- 107 = Excessive Propane content.
- 109 = Excessive Butane content.
- 110 = Excessive Water content.
- 111 = Excessive Pentane content.
- 115 = Excessive Argon content.
- 116 = Excessive Oxygen content.
- 117 = Insufficient Methane content.
- 118 = Excessive Nitrogen content.
- 119 = Excessive Carbon Dioxide content.
- 120 = Excessive Ethane content.
- 130 = Real Specific Gravity value required.
- 131 = Invalid Specific Gravity value.
- 132 = Invalid Heating Value value.
- 133 = Sum of mole fractions does not equal 1.
- 140 = Negative square root (eqn. 5-15, pg27 AGA8 2017).
- 141 = Invalid molar density (*i.e.*,  $d \leq 0$ ) (pg28 AGA8 2017).
- 142 = Invalid H<sub>CH</sub> (eqn. 5-9, pg26 AGA8 2017).
- 143 = Excessive Heptane content.
- 144 = Excessive Octane content.



- 145 = Excessive Nonane content.
- 146 = Excessive Decane content.
- 148 = Error calculating mixture size parameter (eqn. 4-4, pg13 AGA8 2017).

## Licensing

---

Calculation Module: Compressibility  
Additional Required Modules: None

## Notes & Considerations

---

- All passed values must be expressed in the listed dimensional units for the associated variable. Calculated values are returned in the listed dimensional units. The *CalcFact* or *CalcT* functions can be used to obtain the required conversion factor for non-standard dimensional units.
- With the exception of the base pressure value, where passed pressure values are expressed in absolute pressure (Psia) dimensional units, the atmospheric pressure value must be added to the gauge pressure value (Psig) before passing the variable to the calculation function. Conversely, where a returned (calculated) value is expressed in absolute pressure units, the atmospheric pressure must be subtracted from the returned value to obtain the gauge pressure value. The atmospheric pressure value may be calculated using the *CalcPatm* function.
- The general process for using this function is as follows:
  - Collect the required variable values from User-entered values.
  - Convert the entered variable values to the appropriate dimensional units.
  - Call the calculation function.
  - If no error code is returned, use or display the returned value to the User. If an error code is returned, display an appropriate message to the User.
  - Store, print, or use the input values and results as appropriate.
- The *SetLicense* function must be called at least once before attempting to execute this calculation routine.
- The Normal Check option performs a check using the “Normal Limits” for the AGA 8 - 1992 Method. The Expanded Check option performs a check using the “Expanded Limits” for the AGA 8 - 1992 Method. For all other methods, the Normal and Expanded Check options perform the same (normal) limits check.

- If using Calculation Methods (calc\_method) 1, 14, 24, or 29:
  - The Ethylene component, pp[1], is added to the Ethane component, pp[2], and the Air component, pp[13], is added to the Nitrogen component, pp[21]. If both the Ethylene and Air components are 0%, then disregard this bullet point.
- If using Calculation Method (calc\_method) 29:
  - The Isobutane and Isopentane components are not used and, instead, are set equal to the n-Butane and n-Pentane components, respectively. That is, Isobutane is set to n-Butane, and Isopentane is set to n-Pentane. International Standard ISO 6976 2016 does not contain relevant data on the components Isobutane and Isopentane, so use caution if using either of these components while using Calculation Method 29. If both the Isopentane and Isobutane components are 0%, then disregard this bullet point.
- See the appropriate GASCalc Calculation Reference for documentation and references for the calculations performed by this function.

## SetLicense - Set License Function

---

The *SetLicense* function verifies the license information and enables the licensed calculation modules.

### Function Syntax

---

```
SetLicense
(
    long regno,
    char *licensee,
    char *expdate,
    char *version,
    char *copyright
)
```

*Note...*

\* Indicates a variable passed by reference (address), all other values passed by value. Calculated values will be returned in the associated “by reference” variables.

### Variable Definitions

---

**copyright** = Specifies the copyright owner and notice. A pointer to a 256 character (string) value.

**expdate** = Specifies the License expiration date. A pointer to a 9 character (string) value.

**licensee** = Specifies the registered Licensee’s name. A pointer to a 256 character (string) value.

**regno** = Specifies the Licensee’s registration number.

**version** = Specifies the version number of the API. A pointer to a 15 character (string) value.

### Return Values

---

The function returns a positive long integer value, which represents the number of enabled/licensed calculation modules. Or if the software has not yet been registered, the function returns a value less than zero, which represents the Serial Number for the software.

### **Licensing**

---

Calculation Module: None

Additional Required Modules: None

### **Notes & Considerations**

---

- If the function returns a negative value, the software has not been registered. The return value represents the Serial Number for the license. Send the serial number to the developer/vendor to receive the required Registration Number.
- If the function returns a zero value, the license has expired. Contact the developer/vendor to renew the license.