

The thermodynamics package*

Karl D. Hammond
hammondkd@missouri.edu

2022/04/29

Abstract

A package, `thermodynamics`, is defined that makes typesetting quantities found in thermodynamics texts relatively simple. The commands are flexible and intended to be relatively intuitive. It handles several sets of notation for total, specific, and molar quantities; allows changes between symbols (e.g., A vs. F for Helmholtz free energy); and greatly simplifies the typesetting of symbols and partial derivatives commonly encountered in mixture thermodynamics. Changes of one's notes from one textbook to another can be achieved relatively easily by changing package options.

1 Introduction

The purpose of this package is to simplify the typesetting of equations in thermodynamics, specifically chemical engineering thermodynamics, which are often cumbersome to enter. For example, consider the following equation:

$$d\underline{U} = \left(\frac{\partial \underline{U}}{\partial \underline{S}}\right)_{\underline{V}, \underline{n}} d\underline{S} + \left(\frac{\partial \underline{U}}{\partial \underline{V}}\right)_{\underline{S}, \underline{n}} d\underline{V} + \sum_{i=1}^C \left(\frac{\partial \underline{U}}{\partial n_i}\right)_{\underline{S}, \underline{V}, n_{j \neq i}} dn_i. \quad (1)$$

This equation is pretty basic, and equations like it occur all the time in thermodynamics. Without this package, you might typeset it like this:

```
d\underline{U} =
  \left(\frac{\partial \underline{U}}{\partial \underline{S}}\right)_{\underline{V}, \underline{n}}
  d\underline{S}
+ \left(\frac{\partial \underline{U}}{\partial \underline{V}}\right)_{\underline{S}, \underline{n}}
  d\underline{V}
+ \sum_{i=1}^C \left(\frac{\partial \underline{U}}{\partial n_i}\right)_{\underline{S}, \underline{V}, n_{j \neq i}}
  dn_i.
```

This is a lot of code, and even then the output is slightly clunky:

$$d\underline{U} = \left(\frac{\partial \underline{U}}{\partial \underline{S}}\right)_{\underline{V}, \underline{n}} d\underline{S} + \left(\frac{\partial \underline{U}}{\partial \underline{V}}\right)_{\underline{S}, \underline{n}} d\underline{V} + \sum_{i=1}^C \left(\frac{\partial \underline{U}}{\partial n_i}\right)_{\underline{S}, \underline{V}, n_{j \neq i}} dn_i.$$

*This document corresponds to `thermodynamics` v1.00, dated 2022/04/29.

It is also frustratingly difficult to change one's notes or handouts from one textbook that uses, say, n_1 to denote moles of component 1 to another textbook that uses N_1 for the same quantity, or perhaps denotes the total internal energy as U or U^t rather than \underline{U} . For example, if you wanted it to be

$$dU = \left(\frac{\partial U}{\partial S}\right)_{V, N_1, \dots, N_n} dS + \left(\frac{\partial U}{\partial V}\right)_{S, N_1, \dots, N_n} dV + \sum_{i=1}^n \left(\frac{\partial U}{\partial N_i}\right)_{S, V, N_1, \dots, [N_i], \dots, N_n} dN_i$$

without changing any of your code—to update it across all handouts, exams, and homework sets after changing textbooks, say—you would be out of luck (or in for a lot of work).

With this package, you could reduce the code to typeset this equation to

```
\begin{equation}
  d\Ut = \Partial*\Ut{\St}{\Vt, \allNs} d\St
  + \Partial*\Ut{\Vt}{\St, \allNs} d\Vt
  + \sumall_i \Partial*\Ut{\Nt_i}{\St, \Vt, \allNsbut{i}} d\Nt_i
\end{equation}
```

and it will render similarly to Equation (1), including the shortened underscores and negative kerning. If you later decide to change the notation such that extensive properties are not underlined, you can do that without changing any of your code (just change a package option). Similarly, if you want \vec{n} replaced by n_1, \dots, n_C , you can do that with a package option, too.

The package handles second derivatives, too. For example,

```
\Partial*\Hm{T}{P} = T\Partial*\Sm{T}{P}
                  = -T\PartialSecond*\Gm{T}{P} = \cP
```

renders (using the default options)

$$\left(\frac{\partial H}{\partial T}\right)_P = T \left(\frac{\partial S}{\partial T}\right)_P = -T \left(\frac{\partial^2 G}{\partial T^2}\right)_P = C_P$$

Similarly, macros are defined for mixed second partial derivatives that allow things like

```
\Partial*\Gpm_i{P}{T, \allNs}
  = \PartialMixSecond*\Gt{P}{\Nt_i}{T, \allNsbut{i}}
  = \PartialMixSecond*\Gt{\Nt_i}{P}{T, \allNsbut{i}}
  = \Partial*\Vt{\Nt_i}{T, P, \allNsbut{i}} = \Vpm_i
```

which renders

$$\left(\frac{\partial \bar{G}_i}{\partial P}\right)_{T, \vec{n}} = \left(\frac{\partial^2 \underline{G}}{\partial P \partial n_i}\right)_{T, n_{j \neq i}} = \left(\frac{\partial^2 \underline{G}}{\partial n_i \partial P}\right)_{T, n_{j \neq i}} = \left(\frac{\partial \underline{V}}{\partial n_i}\right)_{T, P, n_{j \neq i}} = \bar{V}_i$$

using the defaults.

2 Using the Package

There are three categories of macros defined in this package: macros that produce symbols (or groups of them), macros that typeset derivatives, and macros that are used internally that the user need not know about. There are also several environments that allow the user to switch locally between different delimiters on partial derivatives.

2.1 Predefined Symbols

The macros used to produce symbols fall into five categories: extensive properties, molar properties, specific properties (i.e., per unit mass), partial molar properties, and shortcut macros (e.g., macros for the heat capacities, saturation pressure, and so forth). The macros corresponding to extensive, molar, and specific properties are shown in Table 1. Examples using the enthalpy are

```
\Ht
\Hm    \[ \Ht \quad \Hm \quad \Hs \quad \Hpm_i. \]
\Hs
```

\Hpm Using the default package options, the above renders as

$$\underline{H} \quad H \quad \hat{H} \quad \overline{H}_i.$$

In addition, the properties in Table 2 are defined for convenience.

```
\Ut    How these symbols are rendered can be customized by package options. As long
\Um    as the user consistently uses \Ut to render the total internal energy, \Um to render the
        molar internal energy, and so forth, switching notation from, say,  $\underline{U}$  to  $U^t$  for extensive
        properties is trivial.
```

```
\cPt   The heat capacities (see Table 2) are generally assumed to be molar (e.g., \cP
\cVt   is interpreted to be the molar heat capacity). To get the specific heat capacities,
\cPs   the macros \cPs and \cVs are provided, which by default render as  $\hat{C}_p$  and  $\hat{C}_V$ ,
\cVs   respectively. There are also extensive versions, so \cPt and \cVt will render as
\cPpm   $\underline{C}_p$  and  $\underline{C}_V$ , respectively. Note that \cP and friends require you to surround \text
\cVpm  with brackets if the portion with \text in it is not the last argument. For example,
        \cP_i^\text{A} will work as expected, but \cP^\text{A}_i will not: you need to
        use \cP^\text{A}_i. Using \cP^\IG_i will work as expected.
```

You can also get partial molar heat capacities via \cPpm and \cVpm, though the latter's mathematical definition is a bit hard to wrap one's head around:

```
\[ \cVpm_i = \Partial{\cVt}{\Nt_i}{T,P,\allNsbut{i}}
        = \frac{\partial}{\partial \Nt_i}
        \left[ -T \PartialSecond{\Ft}{T}{\Vt,\allNs}
        \right]_{T,P,\allNsbut{i}} \]
```

yields

$$\overline{C}_{V,i} = \left(\frac{\partial C_V}{\partial n_i} \right)_{T,P,n_{j \neq i}} = \frac{\partial}{\partial n_i} \left[-T \left(\frac{\partial^2 A}{\partial T^2} \right)_{V,\vec{n}} \right]_{T,P,n_{j \neq i}}$$

2.2 Partial Molar Properties

```
\Upm   Partial molar quantities are typically defined with the suffix pm. For example, \Upm
\Hpm   refers to the partial molar internal energy. There are two options for how to enter
        partial molar quantities: as commands or as super/subscripts. For example,
```

```
\[ \Upm{i} \quad \Upm[\IG]{i} \quad \Hpm_i \quad \Hpm^\IG_i \]
```

will typeset as

$$\overline{U}_i \quad \overline{U}_i^{\text{IG}} \quad \overline{H}_i \quad \overline{H}_i^{\text{IG}}$$

There are also partial molar heat capacities available via \cPpm and \cVpm. **Important:** The \text command defined by the amstext package is usually robust enough

Table 1. Commands defined in this package to represent extensive thermodynamic quantities and their molar and specific analogs. These macros should be used even if the symbol the user wishes to use does not match the command used (e.g., `\Ft` for total Helmholtz free energy even if it ends up being set as \underline{A}).

Property	Total	Molar	Specific	Partial Molar
Heat	<code>\Qt</code>	<code>\Qm</code>	<code>\Qs</code>	N/A
Work	<code>\Wt</code>	<code>\Wm</code>	<code>\Ws</code>	N/A
Total energy	<code>\Et</code>	<code>\Em</code>	<code>\Es</code>	<code>\Epm</code>
Internal energy	<code>\Ut</code>	<code>\Um</code>	<code>\Us</code>	<code>\Upm</code>
Enthalpy	<code>\Ht</code>	<code>\Hm</code>	<code>\Hs</code>	<code>\Hpm</code>
Entropy	<code>\St</code>	<code>\Sm</code>	<code>\Ss</code>	<code>\Spm</code>
Volume	<code>\Vt</code>	<code>\Vm</code>	<code>\Vs</code>	<code>\Vpm</code>
Helmholtz free energy	<code>\Ft</code>	<code>\Fm</code>	<code>\Fs</code>	<code>\Fpm</code>
Gibbs free energy	<code>\Gt</code>	<code>\Gm</code>	<code>\Gs</code>	<code>\Gpm</code>
Surface area	<code>\At</code>	<code>\Am</code>	<code>\As</code>	<code>\Apm</code>
Grand potential ^a	<code>\Lt</code>	<code>\Lm</code>	<code>\Ls</code>	<code>\Lpm</code>
Moles	<code>\Nt</code>	N/A	N/A	N/A
B (generic property)	<code>\Bt</code>	<code>\Bm</code>	<code>\Bs</code>	<code>\Bpm</code>
M (generic property)	<code>\Mt</code>	<code>\Mm</code>	<code>\Ms</code>	<code>\Mpm</code>

^aThe grand potential, $\underline{\Omega}(T, \underline{V}, \underline{\mu}) = \underline{U} - T\underline{S} - \sum_i^C \mu_i n_i$, is also called the Landau free energy by some authors.

that something like `\Um^{\text{L}}` will work as expected, without additional braces. This does *not* work for partial molar properties; for example, `\Hpm^{\text{L}}_i` will produce an error, as will `\Hpm_i^{\text{L}}`. The expression `\Hpm_i^{\text{L}}` will work as expected.

`\partialmolar` New partial molar properties can be defined for any “simple” symbol using the `\partialmolar` macro. “Simple” means it has no subscripts or superscripts. For example, the macro for the partial molar Gibbs free energy is defined via the macro

```
\newcommand*{\Gpm}{\partialmolar{Gibbs@symbol}}
```

A list of pre-defined macros for total, molar, specific, and partial molar quantities commonly used in thermodynamics is included in Table 1.

2.3 Other Predefined Symbols and Modifiers

There are a number of predefined symbols and modifiers. While these symbols could be defined or used without these macros, such use is not recommended: changing package options will result in inconsistencies if these macros are not used.

2.3.1 Heat Capacities, Compressibilities, and Expansivities

`\cP` The isobaric and isochoric heat capacities are produced with `\cP` and `\cV`, respectively. Four other measurable quantities are defined: the isothermal and isentropic compressibilities, `\kappaT` and `\kappaS`, respectively; and the isobaric and isentropic volume expansivities, `\alphaP` and `\alphaS`, respectively. Some textbooks use β instead of α for the volume expansivity to differentiate it from the *linear* expansivity; `\alphaP` `\alphaS`

Table 2. Convenience macros and their default symbols. These are generally “smart”: for example, `\cP_i` renders as $C_{P,i}$, as expected, and `\cP_i^\circ` renders as $C_{P,i}^\circ$, also as expected. You can also reverse it: `\cP^\circ_i` becomes $C_{P,i}^\circ$.

Name	Macro	Sym.	Definition	Base Symbol Macro
Isobaric heat capacity	<code>\cP^a</code>	C_P	$T \left(\frac{\partial S}{\partial T} \right)_P$	<code>\heatcapacitysymbol</code>
Isochoric heat capacity	<code>\cV^a</code>	C_V	$T \left(\frac{\partial S}{\partial T} \right)_V$	<code>\heatcapacitysymbol</code>
Isothermal compressibility	<code>\kappa_T</code>	κ_T	$-\frac{1}{V} \left(\frac{\partial V}{\partial P} \right)_T$	<code>\compressibilitysymbol</code>
Isentropic compressibility	<code>\kappa_S</code>	κ_S	$-\frac{1}{V} \left(\frac{\partial V}{\partial P} \right)_S$	<code>\compressibilitysymbol</code>
Isobaric expansivity	<code>\alpha_P</code>	α_P	$\frac{1}{V} \left(\frac{\partial V}{\partial T} \right)_P$	<code>\expansivitysymbol</code>
Isentropic expansivity	<code>\alpha_S</code>	α_S	$\frac{1}{V} \left(\frac{\partial V}{\partial T} \right)_S$	<code>\expansivitysymbol</code>
Pure fugacity	<code>\fpure</code>	f	ϕP	
Mixture fugacity	<code>\fmix</code>	\hat{f}	$\hat{f}_i = x_i \hat{\phi}_i P$	
Saturation fugacity	<code>\fsat</code>	f^{sat}	$\phi^{\text{sat}} P^{\text{sat}}$	<code>\sat</code>
Pure fugacity coeff.	<code>\phipure</code>	ϕ	$\phi_i = \exp \left(\frac{1}{RT} \int_0^P V_i(T, p) - \frac{RT}{p} dp \right)$	
Mixture fugacity coeff.	<code>\phimix</code>	$\hat{\phi}$	$\hat{\phi}_i = \exp \left(\frac{1}{RT} \int_0^P \bar{V}_i(T, p, \bar{x}) - \frac{RT}{p} dp \right)$	
Henry’s constant (rational)	<code>\Henryrat</code>	h	$\gamma_i^\infty f_i$	
Henry’s constant (molal)	<code>\Henrymol</code>	\mathcal{H}	$M_s \gamma_i^\infty f_i$	
Rational activity coeff.	<code>\gamarat</code>	γ^*	γ / γ^∞	
Molal activity coeff.	<code>\gammamol</code>	γ^\square	$x_s \gamma / \gamma^\infty$	
Saturation fugacity coeff.	<code>\phisat</code>	ϕ^{sat}	$\phi(T, P^{\text{sat}})$	<code>\sat</code>
Saturation pressure	<code>\Psat</code>	P^{sat}		<code>\sat</code>
Vapor pressure	<code>\Pvap</code>	Currently a synonym for <code>\Psat</code> ^b		
Standard state	<code>\std^c</code>	\circ		
Standard pressure	<code>\Pstd</code>	P°		<code>\std</code>
Standard fugacity	<code>\fstd</code>	f°	$f(T, P^\circ)$	<code>\std</code>
Change on mixing	<code>\Deltamix^d</code>	ΔM_{mix}	$M - \sum_i x_i M_i$	<code>\mixing</code>
Change on reaction	<code>\Deltarxn^d</code>	ΔM_{rxn}	$\sum_i \nu_i M_i$	<code>\reaction</code>
Change on melting	<code>\Deltafus^d</code>	ΔM^{fus}	$M^L - M^S$	<code>\fusion</code>
Change on boiling	<code>\Deltavap^d</code>	ΔM^{vap}	$M^V - M^L$	<code>\vaporization</code>
Change on subliming	<code>\Deltasub^d</code>	ΔM^{sub}	$M^V - M^S$	<code>\sublimation</code>

^aExtensive and specific (per-unit-mass) versions are available as `\cPt` and `\cPs`, respectively, with similar macros for the isochoric heat capacity.

^bIf you want `\Pvap` to produce P^{vap} instead of P^{sat} , you should redefine the `\sat` macro.

^cTypical usage would be $\mu_i = \mu_i^\circ + RT \log a_i$, yielding $\mu_i = \mu_i^\circ + RT \log a_i$.

^dThe usual usage would be something like $\Delta M_{\text{mix}}^{\text{VM}} = 0$.

this can be changed by redefining `\expansivitysymbol`, which is done automatically by some of the package options that create notation specific to a particular textbook.

2.3.2 Fugacities and Fugacity Coefficients

`\fpure` Different textbooks use different modifications of the symbol f for fugacity, so it is recommended to use the macro `\fpure` to denote the pure-component fugacity and `\fmix` to denote the mixture fugacity. Similarly, the pure-component fugacity coefficient should be generated with `\phipure`, and that in the mixture should be `\phimix`.

For example, the following markup is an example of a common equation in mixture thermodynamics:

```
\[ \fmix_j = x_j \phimix_j P = x_j \gamma_j \fpure_j. \]
```

With the default package options, this produces

$$\hat{f}_j = x_j \hat{\phi}_j P = x_j \gamma_j f_j.$$

With the Thompson package option, however, the same markup produces

$$\hat{f}_j = x_j \hat{\phi}_j P = x_j \gamma_j f_j^\bullet.$$

Similarly, the Prausnitz package option causes it to generate

$$f_j = x_j \phi_j P = x_j \gamma_j f_{\text{pure},j},$$

and the Sandler option causes it to generate

$$\bar{f}_j = x_j \bar{\phi}_j P = x_j \gamma_j f_j.$$

2.3.3 Activity Coefficients and Henry's Constants

`\gammarat` The activity coefficient can be generated with `\gamma`, as usual. The Henry's Law activity coefficients should be produced with `\gammarat` (rational basis) and `\gammamol` (molal basis). There are also macros to generate the Henry's law constants for both the rational basis (`\Henryrat`) and the molal basis (`\Henrymol`). These are interrelated:

```
\[ \fmix_i = x_i \gamma_i \fpure_i = x_i \gammarat_i \Henryrat_i
      = C_i \gammamol_i \Henrymol_i \]
```

produces

$$\hat{f}_i = x_i \gamma_i f_i = x_i \gamma_i^* h_i = C_i \gamma_i^\square \mathcal{H}_i$$

using the default options.

2.3.4 Saturation Properties

`\Psat` The saturation pressure is generated with `\Psat`. The macro `\Pvap` is an alias for `\Psat`. The fugacity and fugacity coefficient at saturation are accessed via `\fsat` and `\phisat`, respectively. Package options can be used to change some of these to match the notation of specific textbooks.

`\sat` The `\sat` macro is used "behind the scenes" as part of `\Psat`, `\fsat`, and `\phisat`, which produce P^{sat} , f^{sat} , and ϕ^{sat} , respectively. If you wanted to redefine them to be P^{vap} , f^{vap} , and ϕ^{vap} , you could simply redefine `\sat` with `\renewcommand*\sat{\text{vap}}`. This is done automatically using package option Sandler.

2.3.5 Standard States

`\std` The symbol \circ (`\circ`) is used by default for standard states. This is intended to be easy to change should the user want to replace P° with P^\ominus , say. This is accessed via the `\Pstd` macro. The macro `\Pstd` is defined as `P^\std` for convenience to denote standard pressures, and `\fstd` is defined for standard fugacities so as to ease implementation across textbooks.

The usual usage would be something like this:

```
\[ \Deltarxn\Gm = \sumall_i \nu_i \mu_i = \sumall_i \left[ \nu_i \mu^\std_i
+ \nu_i RT \log\left(\frac{f_{mix_i}}{f^\std_i}\right) \right]
= \Deltarxn\Gm^\std + RT \log\left[\prodall_i a_i^{\nu_i} \right] \]
```

which produces

$$\Delta G_{\text{rxn}} = \sum_{i=1}^C \nu_i \mu_i = \sum_{i=1}^C \left[\nu_i \mu_i^\circ + \nu_i RT \log \left(\frac{f_i}{f_i^\circ} \right) \right] = \Delta G_{\text{rxn}}^\circ + RT \log \left[\prod_{i=1}^C a_i^{\nu_i} \right]$$

with the default settings.

2.3.6 Changes on Mixing, Reaction, Fusion, Vaporization, and Sublimation

`\Deltamix` Mixing properties are handled via the `\Deltamix` macro, and are used as in the following example:

```
\[ \Deltamix\Gm = \Gm - \sumall_i x_i \Gm_i = \Deltamix\Hm - T\Deltamix\Sm \]
```

which yields

$$\Delta G_{\text{mix}} = G - \sum_{i=1}^C x_i G_i = \Delta H_{\text{mix}} - T \Delta S_{\text{mix}}.$$

`\mixing` Some textbooks (Sandler, Thompson) choose to typeset these with the word “mix” before the symbol, which is handled automatically by this package. The macro `\mixing` determines how the change in mixing label is rendered; the default is `\text{mix}`.

`\Deltafus` The commands `\Deltafus`, `\Deltasub`, and `\Deltavap` typeset changes due to fusion (melting), sublimation (subliming), and vaporization (boiling), respectively.
`\Deltasub`
`\Deltavap` Their use is straightforward, viz.,

```
\[ \Deltasub\Hm = \Hm^V - \Hm^S = \Hm^V - \Hm^L + (\Hm^L - \Hm^S)
= \Deltafus\Hm + \Deltavap\Hm \]
```

yielding

$$\Delta H^{\text{sub}} = H^V - H^S = H^V - H^L + (H^L - H^S) = \Delta H^{\text{vap}} + \Delta H^{\text{fus}}$$

with the default options. Note that some textbooks (e.g., Sandler) typeset these quantities quite differently.

`\Deltaf` The macros `\Deltaf` and `\Deltarxn` are intended to typeset the enthalpy or free energy of formation and reaction, respectively. For example,

```
\[ \Deltarxn\Hm^\std = \sumall_i \nu_i \Deltaf\Hm_i^\std \]
```

results in

$$\Delta H_{\text{rxn}}^{\circ} = \sum_{i=1}^C \nu_i \Delta H_{f,i}^{\circ}.$$

It is not anticipated that this command will be combined with something like a heat capacity, which already has a (potentially double) subscript, but as there is no “formation” heat capacity, that should not present a problem.

2.4 Residual and Excess Properties

`\UR` Additional macros are defined that make it easy to typeset the residual (also called
`\URt` “departure”) and excess total, molar, specific, and partial molar properties. These
`\URs` macros follow the same pattern: `\UR`, `\URt`, `\URs`, and `\URpm` typeset the molar, total,
`\URpm` specific, and partial molar residual internal energies, respectively, and by default
`\UE` expand to U^R , \underline{U}^R , \hat{U}^R , and \overline{U}_i^R (the last is called as `\URpm{i}` or `\URpm_i`). Similarly,
`\UET` `\UE`, `\UET`, `\UES`, and `\UEpm` typeset the corresponding excess properties. The first
`\UES` character of the macros for other properties follow the same pattern as in Table 1.
`\UEpm` The R and E characters are generated by the macros `\residual` and `\excess`,
`\excess` respectively. These macros can be redefined; for example, if you want `\SE`, which
`\residual` normally produces S^E , to give you S^{EX} —and let’s be honest, who doesn’t want that?¹—
then you can redefine it with

```
\renewcommand*\excess{{EX}}
```

or possibly

```
\renewcommand*\excess{{\mathrm{EX}}}
```

or even

```
\renewcommand*\excess{\text{EX}}
```

which cause `\SE` to expand to S^{EX} , S^{EX} , and S^{EX} , respectively.

It is generally not possible to use superscripts with the excess or residual properties; in the event the user needs this, the `\excess` and `\residual` macros can be used directly, viz.,

```
\begin{gather*}
\Hm^{\excess,\std} = \HE(T,\Pstd) \\\
\renewcommand*\excess{EX}
\Hm^{\excess,\std} = \HE(T,\Pstd)
\end{gather*}
```

which yields

$$\begin{aligned} H^{E,\circ} &= H^E(T, P^\circ) \\ H^{EX,\circ} &= H^{EX}(T, P^\circ) \end{aligned}$$

using the default options.

¹You knew that joke was coming.

2.5 Partial Derivatives

`\Partial` Partial derivatives are easily rendered using the `\Partial` command. There is a starred form (`\Partial*`) that additionally adjusts the spacing after the closing symbol to remove some of the space, anticipating that the following binary operator will overhang the subscripts. Compare the following:

```
\begin{gather*}
\Partial{\Hm}{T}{P} = \cP \quad \Partial*{\Hm}{T}{P} = \cP
\end{gather*}
```

which yields

$$\left(\frac{\partial H}{\partial T}\right)_P = C_P \quad \left(\frac{\partial H}{\partial T}\right)_P = C_P$$

`\PartialSecond` Second partial derivatives and mixed-second partial derivatives are typeset with the commands `\PartialSecond` and `\PartialMixSecond`, respectively. Like the first-order variety, these also have starred versions that remove the space immediately following the closing symbols, anticipating that the equals sign or other binary operator following the derivative will overhang the elements held constant. For example,

```
\[ \Vpm_i = \Partial*{\Vt}{\Nt_i}{T,P,\allNsbut{i}}
= \PartialMixSecond*{\Gt}{\Nt_i}{P}{T,\allNsbut{i}}
= \PartialMixSecond*{\Gt}{P}{\Nt_i}{T,\allNsbut{i}}
= \Partial{\Gpm_i}{P}{T,\allNs} \]
```

looks like

$$\bar{V}_i = \left(\frac{\partial V}{\partial n_i}\right)_{T,P,n_{j\neq i}} = \left(\frac{\partial^2 G}{\partial n_i \partial P}\right)_{T,n_{j\neq i}} = \left(\frac{\partial^2 G}{\partial P \partial n_i}\right)_{T,n_{j\neq i}} = \left(\frac{\partial \bar{G}_i}{\partial P}\right)_{T,\vec{n}}$$

`\PartialBigg` There are instances (such as the line above) when `\Partial` causes parentheses that are slightly too tall but do not need to be—particularly when partial molar properties, specific quantities, or fugacities are involved. The macro `\PartialBigg` uses `amsmath`'s `\Biggl` and `\Biggr` macros in place of `\left` and `\right` to size the parentheses accordingly; `\Partialbigg` uses `\biggl` and `\biggr` in a similar fashion. For example, compare the following:

```
\[ \Vpm_i = \Partial*{\Vt}{\Nt_i}{T,P,\allNsbut{i}}
= \Partial*{\Gpm_i}{P}{T,\allNs}
= \PartialBigg*{\Gpm_i}{P}{T,\allNs}
= RT\Partial*{\log\fmix_i}{P}{T,\allNs}
= RT\,\PartialBigg*{\log\fmix_i}{P}{T,\allNs}
= RT\,\Partialbigg{\log\fmix_i}{P}{T,\allNs} \]
```

which typesets as

$$\bar{V}_i = \left(\frac{\partial V}{\partial n_i}\right)_{T,P,n_{j\neq i}} = \left(\frac{\partial \bar{G}_i}{\partial P}\right)_{T,\vec{n}} = \left(\frac{\partial \bar{G}_i}{\partial P}\right)_{T,\vec{n}} = RT \left(\frac{\partial \log \hat{f}_i}{\partial P}\right)_{T,\vec{n}} = RT \left(\frac{\partial \log \hat{f}_i}{\partial P}\right)_{T,\vec{n}} = RT \left(\frac{\partial \log \hat{f}_i}{\partial P}\right)_{T,\vec{n}}$$

Note that a similar effect—possibly with other side effects—can be achieved with `amsmath`'s `\smash` command, which has the effect of removing all vertical space associated with a particular character. Observe:

`\[\Vpm_i = \Partial{\Vt}{\Nt_i}{T,P,\allNsbut{i}}
= \Partial{\smash{\Gpm_i}}{P}{T,\allNs} \]`

produces

$$\bar{V}_i = \left(\frac{\partial V}{\partial n_i} \right)_{T,P,n_{j \neq i}} = \left(\frac{\partial \bar{G}_i}{\partial P} \right)_{T,\bar{n}} .$$

Note that the vertical spacing is not quite as good here as it was above. This can be fixed by using the optional argument to `\smash`, viz.,

`\[\Vpm_i = \Partial{\Vt}{\Nt_i}{T,P,\allNsbut{i}}
= \Partial{\smash[t]{\Gpm_i}}{P}{T,\allNs} \]`

$$\bar{V}_i = \left(\frac{\partial V}{\partial n_i} \right)_{T,P,n_{j \neq i}} = \left(\frac{\partial \bar{G}_i}{\partial P} \right)_{T,\bar{n}} .$$

It is possible to “fake” higher-order derivatives via some trickery. For example,

`\[\cPpm_i = T\Partial{\Spm_i}{T}{P,\allXs}
= T\PartialSecond{\St}{T}{\Nt_i}{P,\allNsbut{i}}
= -T\Partial{\^3\Gt}{T^2\partial\Nt_i}{P,\allNsbut{i}} \]`

gives

$$\bar{C}_{P,i} = T \left(\frac{\partial \bar{S}_i}{\partial T} \right)_{P,\bar{x}} = T \left(\frac{\partial^2 S}{\partial T \partial n_i} \right)_{P,n_{j \neq i}} = -T \left(\frac{\partial^3 G}{\partial T^2 \partial n_i} \right)_{P,n_{j \neq i}} ,$$

which is probably pretty close to what you wanted. Using this trickery with the package option `nosubscripts` will not work as well, and the use of third- and higher-order derivatives with this package should generally be considered unsupported.

2.6 Holding Constant the Number of Moles of Several Species

`\allNs` It is common in thermodynamics to use notation such as

`\allNsbut
\allmus
\allmusbut
\allXs`

$$\bar{V}_k = \left(\frac{\partial V}{\partial n_k} \right)_{T,P,n_{j \neq k}}$$

`\allXsbut` or perhaps

`\allYs
\allYsbut`

$$\bar{V}_k = \left(\frac{\partial V}{\partial n_k} \right)_{T,P,n_1,\dots,[n_k],\dots,n_C}$$

`\allMs
\allMsbut
\allWs
\allWsbut`

to mean partial derivatives that hold the number of moles of each species constant *except* the one being changed. Similarly, a property determined with all mole fractions held constant might be written

$$C_P = \left(\frac{\partial H}{\partial T} \right)_{P,\bar{x}}$$

or perhaps

$$C_P = \left(\frac{\partial H}{\partial T} \right)_{P,x_1,\dots,x_C} .$$

Table 3. Options controlling which symbols to use by default. The macros `\Et`, `\Ut`, `\Ft`, `\Gt`, `\Ht`, `\At`, and `\Nt` represent the total energy, internal energy, Helmholtz free energy, Gibbs free energy, enthalpy, surface area, and number of moles, respectively. Symbols are shown as they would appear with the (default) option `intensive-plain`.

Option	<code>\Et</code>	<code>\Ut</code>	<code>\Ft</code>	<code>\Gt</code>	<code>\Ht</code>	<code>\At</code>	<code>\Nt</code>
<code>EUAGHan</code>	\underline{E}	\underline{U}	\underline{A}	\underline{G}	\underline{H}	\underline{a}	\underline{n}
<code>EUAGHaN</code>	$\underline{\underline{E}}$	$\underline{\underline{U}}$	$\underline{\underline{A}}$	$\underline{\underline{G}}$	$\underline{\underline{H}}$	$\underline{\underline{a}}$	$\underline{\underline{N}}$
<code>EUHAGan</code>	(synonym for <code>EUAGHan</code>)						
<code>EUHAGaN</code>	(synonym for <code>EUAGHaN</code>)						
<code>EUFGHan</code>	\underline{E}	\underline{U}	\underline{F}	\underline{G}	\underline{H}	\underline{A}	\underline{n}
<code>EUFGHAN</code>	$\underline{\underline{E}}$	$\underline{\underline{U}}$	$\underline{\underline{F}}$	$\underline{\underline{G}}$	$\underline{\underline{H}}$	$\underline{\underline{A}}$	$\underline{\underline{N}}$
<code>EEFGHan</code>	$\underline{\underline{E}}$	$\underline{\underline{E}}$	$\underline{\underline{F}}$	$\underline{\underline{G}}$	$\underline{\underline{H}}$	$\underline{\underline{A}}$	$\underline{\underline{n}}$
<code>EEFGHAN</code>	$\underline{\underline{\underline{E}}}$	$\underline{\underline{\underline{E}}}$	$\underline{\underline{\underline{F}}}$	$\underline{\underline{\underline{G}}}$	$\underline{\underline{\underline{H}}}$	$\underline{\underline{\underline{A}}}$	$\underline{\underline{\underline{N}}}$
<code>EEFGHan</code>	$\underline{\underline{E}}$	$\underline{\underline{E}}$	$\underline{\underline{F}}$	$\underline{\underline{G}}$	$\underline{\underline{H}}$	$\underline{\underline{a}}$	$\underline{\underline{n}}$
<code>EEFGHaN</code>	$\underline{\underline{\underline{E}}}$	$\underline{\underline{\underline{E}}}$	$\underline{\underline{\underline{F}}}$	$\underline{\underline{\underline{G}}}$	$\underline{\underline{\underline{H}}}$	$\underline{\underline{\underline{a}}}$	$\underline{\underline{\underline{N}}}$
<code>EEAGHaN</code>	$\underline{\underline{E}}$	$\underline{\underline{E}}$	$\underline{\underline{A}}$	$\underline{\underline{G}}$	$\underline{\underline{H}}$	$\underline{\underline{a}}$	$\underline{\underline{N}}$
<code>EUAGHan</code>	$\underline{\underline{E}}$	$\underline{\underline{U}}$	$\underline{\underline{A}}$	$\underline{\underline{G}}$	$\underline{\underline{H}}$	$\underline{\underline{A}}$	$\underline{\underline{n}}$
<code>EUAGHAN</code>	$\underline{\underline{\underline{E}}}$	$\underline{\underline{\underline{U}}}$	$\underline{\underline{\underline{A}}}$	$\underline{\underline{\underline{G}}}$	$\underline{\underline{\underline{H}}}$	$\underline{\underline{\underline{A}}}$	$\underline{\underline{\underline{N}}}$
<code>EUFGHan</code>	$\underline{\underline{E}}$	$\underline{\underline{U}}$	$\underline{\underline{F}}$	$\underline{\underline{G}}$	$\underline{\underline{H}}$	$\underline{\underline{a}}$	$\underline{\underline{n}}$
<code>EUFGHaN</code>	$\underline{\underline{\underline{E}}}$	$\underline{\underline{\underline{U}}}$	$\underline{\underline{\underline{F}}}$	$\underline{\underline{\underline{G}}}$	$\underline{\underline{\underline{H}}}$	$\underline{\underline{\underline{a}}}$	$\underline{\underline{\underline{N}}}$

3 Loading the Package

To load the package with the defaults enabled, load it the usual way:

```
\usepackage{thermodynamics}
```

The package options loaded by default are `EUAGHan`, `subscripts`, `parentheses`, `intensive-plain`, and `moles-index`. These define, respectively, the default symbols to use for total energy, internal energy, Helmholtz free energy, and so forth; the manner of writing partial derivatives; the delimiters around partial derivatives; and the manner of denoting extensive, molar, and specific properties. The default behavior can be altered by options in the following section.

3.1 Package Options

There are three categories of options: options that affect which symbols are used, options that affect how symbols are decorated, and options that affect how partial derivatives are displayed. These are presented in turn.

3.1.1 Options that Change Symbol Sets

There are several options that choose the set of symbols to use for total energy, internal energy, Helmholtz free energy, and so forth. These are summarized in Table 3. The default is `EUAGHan`.

Using `EUAGHan` (the default), we might use the following markup:

```
\[ \Ft = \Ut - T\St = -P\Vt + \sum_i \mu_i \Nt_i + \sigma d\At
\quad \Hm = \Um + P\Vm \quad \Et = \Ut + \frac{1}{2} mv^2 \]
```

which would look like

$$\underline{A} = \underline{U} - T\underline{S} = -P\underline{V} + \sum_i \mu_i n_i + \sigma d\underline{a} \quad H = U + PV \quad \underline{E} = \underline{U} + \frac{1}{2}mv^2.$$

Using the EUGHAN option, the same markup would yield

$$\underline{F} = \underline{U} - T\underline{S} = -P\underline{V} + \sum_i \mu_i N_i + \sigma d\underline{A} \quad H = U + PV \quad \underline{E} = \underline{U} + \frac{1}{2}mv^2.$$

3.1.2 Options for Extensive vs. Molar Properties

There are four sets of notation that define how extensive properties are represented, as shown in Table 4. The default is `intensive-plain`, which (using the volume as an example) represents the total, molar, specific, and partial molar volumes, respectively, as \underline{V} , V , \hat{V} , and \bar{V}_j , respectively.

For example, the definition of the partial molar enthalpy would be different depending on which set of notation is used. The markup

$$\begin{aligned} \backslash\text{Hpm}_i &= \backslash\text{Partial}\{\text{Ht}\}\{\text{Nt}_i\}\{\text{T,P},\backslash\text{allNsbut}\{i\}\} \\ &= \backslash\text{Partial}\{\text{Nt}\backslash\text{Hm}\}\{\text{Nt}_i\}\{\text{T,P},\backslash\text{allNsbut}\{i\}\} \end{aligned}$$

yields the following, depending on the package option loaded:

$$\begin{aligned} \bar{H}_i &= \left(\frac{\partial \underline{H}}{\partial n_i} \right)_{T,P,n_{j \neq i}} = \left(\frac{\partial nH}{\partial n_i} \right)_{T,P,n_{j \neq i}} && \text{intensive-plain} \\ \bar{H}_i &= \left(\frac{\partial H}{\partial n_i} \right)_{T,P,n_{j \neq i}} = \left(\frac{\partial nH}{\partial n_i} \right)_{T,P,n_{j \neq i}} && \text{extensive-plain} \\ \bar{H}_i &= \left(\frac{\partial H^t}{\partial n_i} \right)_{T,P,n_{j \neq i}} = \left(\frac{\partial nH}{\partial n_i} \right)_{T,P,n_{j \neq i}} && \text{extensive-superscript} \\ \bar{H}_i &= \left(\frac{\partial H}{\partial n_i} \right)_{T,P,n_{j \neq i}} = \left(\frac{\partial nh}{\partial n_i} \right)_{T,P,n_{j \neq i}} && \text{intensive-lowercase} \end{aligned}$$

The use of `intensive-lowercase` is strongly discouraged.

Note that the number of moles can be changed from n to N via the options in the previous section.

3.1.3 Options Affecting Partial Derivatives

There are several options that change how partial derivatives are rendered. First are the options that affect the delimiters. We will use the following code as an example:

$$\begin{aligned} \backslash[\backslash\text{Partial}*\{\backslash\text{Vm}\}\{\text{T}\}\{\text{P}\} &= \backslash\text{PartialMixSecond}\{\backslash\text{Gm}\}\{\text{T}\}\{\text{P}\}\{\} \\ &= \backslash\text{PartialMixSecond}\{\backslash\text{Gm}\}\{\text{P}\}\{\text{T}\}\{\} \\ &= -\backslash\text{Partial}\{\backslash\text{Sm}\}\{\text{P}\}\{\text{T}\}. \backslash] \end{aligned}$$

Using the `parentheses` option (the default), this gives

$$\left(\frac{\partial V}{\partial T} \right)_P = \left(\frac{\partial^2 G}{\partial T \partial P} \right) = \left(\frac{\partial^2 G}{\partial P \partial T} \right) = - \left(\frac{\partial S}{\partial P} \right)_T.$$

Table 4. Notation sets that can be set using the options `intensive-plain` (the default), `extensive-plain`, `extensive-superscript`, and `intensive-lowercase`, respectively. Note that specific quantities are generally assumed to look like the molar symbol with a caret on top.

Option	<code>\Vt</code>	<code>\Vm</code>	<code>\Vs</code>	<code>\Vpm_i</code>
<code>intensive-plain</code>	\underline{V}	V	\hat{V}	\bar{V}_i
<code>extensive-plain</code>	V	\underline{V}	\hat{V}	\bar{V}_i
<code>extensive-superscript</code>	V^t	V	\hat{V}	\bar{V}_i
<code>intensive-lowercase</code>	V	v	\hat{v}	\bar{v}_i

The option `brackets` changes the output to

$$\left[\frac{\partial V}{\partial T} \right]_P = \left[\frac{\partial^2 G}{\partial T \partial P} \right] = \left[\frac{\partial^2 G}{\partial P \partial T} \right] = - \left[\frac{\partial S}{\partial P} \right]_T.$$

The option `bar` changes the output to

$$\frac{\partial V}{\partial T} \Big|_P = \frac{\partial^2 G}{\partial T \partial P} = \frac{\partial^2 G}{\partial P \partial T} = - \frac{\partial S}{\partial P} \Big|_T.$$

The option `plain-derivatives` eliminates all delimiters; this forces the `nosubscripts` option. The output in this case is

$$\frac{\partial V(T,P)}{\partial T} = \frac{\partial^2 G(T,P)}{\partial T \partial P} = \frac{\partial^2 G(P,T)}{\partial P \partial T} = - \frac{\partial S(P,T)}{\partial P}$$

Accompanying the `plain-derivatives` option is the `nosubscripts` option, which overrides the default option `subscripts`. This option makes partial derivatives such as

$$\left(\frac{\partial V}{\partial P} \right)_T \quad (\text{subscripts option}),$$

and instead renders them

$$\left(\frac{\partial V(P,T)}{\partial P} \right) \quad (\text{nosubscripts option}).$$

Combined with `plain-derivatives`, this would give

$$\frac{\partial V(P,T)}{\partial P} \quad (\text{nosubscripts and plain-derivatives options}).$$

The order of the variables is determined by the arguments given: it is always written with the variable that is changing first, and the other variables in the order given in the final argument to `\Partial` and friends. This will result in things like the following:

$$\begin{aligned} \backslash [\backslash \text{Partial}\{\backslash \text{Vm}\}\{\text{T}\}\{\text{P}\} &= \backslash \text{PartialMixSecond}\{\backslash \text{Gm}\}\{\text{T}\}\{\text{P}\}\{\} \\ &= \backslash \text{PartialMixSecond}\{\backslash \text{Gm}\}\{\text{P}\}\{\text{T}\}\{\} = -\backslash \text{Partial}\{\backslash \text{Sm}\}\{\text{P}\}\{\text{T}\}, \backslash] \end{aligned}$$

which would produce (using plain-derivatives)

$$\frac{\partial V(T,P)}{\partial T} = \frac{\partial^2 G(T,P)}{\partial T \partial P} = \frac{\partial^2 G(P,T)}{\partial P \partial T} = -\frac{\partial S(P,T)}{\partial P}$$

This is not unclear, but the variables appear in a different order.

`thermoparentheses` If you want to use parentheses *locally*, even though your overall document uses another delimiter, the `thermoparentheses` environment will do that. Similarly, `thermobrackets` will temporarily switch to brackets, `thermobar` will temporarily switch to a trailing vertical bar, and `thermoplain` will remove delimiters altogether.

`thermosubscripts` The environments `thermosubscripts` and `thermoNOsubscripts` force the use or disuse of subscripts, respectively.

`thermoNOsubscripts`

3.1.4 Options Regarding the Number of Moles

`\allNs` The default option `moles-index` defines the macro `\allNs` to expand to \vec{n} and the macro `\allNsbut{i}` to expand to $n_{j \neq i}$. You can change the dummy index: `\allNsbut[k]{i}` expands to $n_{k \neq i}$ by default. This is typically not necessary, however: if you type `\allNsbut{j}`, the package will figure out that you want $n_{k \neq j}$ rather than $n_{j \neq j}$. The time to use the optional argument is in situations such as

$$\left(\frac{\partial \mu_j}{\partial n_k} \right)_{n_{i \neq k}},$$

which is incorrect if the dummy index j is used in place of the i .

You can change these to expand to ranges using the `moles-range` option, which renders `\allNs` as n_1, \dots, n_C and `\allNsbut{i}` as $n_1, \dots, [n_i], \dots, n_C$. The optional argument is ignored in this set of notation. Examples of these options are shown in Table 5.

`\ncomponents` You can change the symbol for the number of components (default: C) by redefining the macro `\ncomponents`.

3.1.5 Other Options

`\dbar` The default for path-dependent one-forms (often called “inexact differentials”) is `\dbar`, which looks like \vec{d} . This can be changed, if desired, to a delta (δ) with the `delta` option to the package.

It should be noted that the `\dbar` macro is not very stable: changing the typeface to something not supported will probably ruin it, as the kerning is very font-specific. This package currently supports Computer Modern, Times, Palatino, Bitstream Charter, Garamond, and Utopia, but other typefaces may require a manual redefinition.

3.1.6 Options for Specific Textbooks

There are several options that load package options and/or redefine particular commands to match the notation in a particular textbook. So far the following textbooks are supported:

Table 5. Illustration of the moles-index and moles-range options and their effects on `\allNs` and `\allNsbut`.

Macro ^a	moles-index	moles-range
<code>\allNs</code>	\vec{n}	n_1, \dots, n_C
<code>\allNsbut{1}</code>	$n_{j \neq 1}$	n_2, \dots, n_C
<code>\allNsbut{i}</code>	$n_{j \neq i}$	$n_1, \dots, [n_i], \dots, n_C$
<code>\allNsbut{j}</code>	$n_{k \neq j}$	$n_1, \dots, [n_j], \dots, n_C$
<code>\allNsbut{\ncomponents}</code>	$n_{j \neq C}$	n_1, \dots, n_{C-1}
<code>\allNsbut[k]{i}</code>	$n_{k \neq i}$	$n_1, \dots, [n_i], \dots, n_C$
<code>\allXs</code>	\vec{x}	x_1, \dots, x_C
<code>\allXsbut{1}</code>	$x_{j \neq 1, C}$	x_2, \dots, x_{C-1}
<code>\allXsbut{i}</code>	$x_{j \neq i, C}$	$x_1, \dots, [x_i], \dots, x_{C-1}$
<code>\allXsbut{j}</code>	$x_{k \neq j, C}$	$x_1, \dots, [x_j], \dots, x_{C-1}$
<code>\allXsbut{\ncomponents-1}</code>	$x_{j \neq C-1, C}$	x_1, \dots, x_{C-2}
<code>\allXsbut[k]{\ncomponents-1}</code>	$x_{k \neq C-1, C}$	x_1, \dots, x_{C-2}
<code>\allXsbut{\ncomponents}</code> ^b	$x_{j \neq C}$	x_1, \dots, x_{C-1}

^aYou may use C directly instead of `\ncomponents` here, but then it will not change $C - 1$ to $C - 2$ (or, say, $M - 1$ to $M - 2$) if you want to redefine `\ncomponents` later.

^bThis would typically be used to denote something like $G(T, P, n, x_1, \dots, x_{C-1})$ rather than in a subscript, but it looks silly if we don't handle this case this way.

Bejan Notation used by Bejan, *Advanced Engineering Thermodynamics*, Third Edition. Wiley: Hoboken, 2006. Loads non-default package options `EUFGHAN`, `intensive-lowercase`, and `delta`. Also swaps the definitions of `\@specific` and `\@intensive` and redefines `\cV`, `\cVs`, `\cVt`, `\expansivitysymbol`, and `\ncomponents` to match his notation.

CBK Notation used by Çengel, Boles, and Kanoglu, *Thermodynamics: An Engineering Approach*, Ninth Edition. McGraw Hill: Singapore, 2020. Loads the non-default package options `EUAGHAN` and `intensive-lowercase`; also redefines partial molar, specific, and molar properties' notation to fit theirs, and redefines `\pressure@symbol`, `\Deltarxn`, `\compressibilitysymbol`, and `\expansivitysymbol` to fit their usage. Their prodigal symbols for specific and total volume are not supported.

ElliottLira Notation used by Elliott and Lira, *Introductory Chemical Engineering Thermodynamics*, Second Edition. Prentice Hall: Upper Saddle River, 2012. Loads the default package options and redefines `\allcomponents` and `\Deltarxn` to fit their notation.

Koretsky Notation used by Koretsky, *Engineering and Chemical Thermodynamics*, Second Edition, Wiley: New Caledonia, 2013. Loads the non-default package options `EUAGHAN`, `brackets`, `intensive-lowercase`, and `delta`; modifies the `intensive-lowercase` defaults to make uppercase partial molar properties to match Koretsky's notation and redefines `\expansivitysymbol`, `\IS`, `\residual`, `\IG`, `\IGM`, `\Deltafus`, `\Deltasub`, `\Deltavap`, `\Henryrat`, `\gammarat`, `\Deltafus`, `\Deltavap`, and `\Deltasub` to match his use. Also redefines the fugacity coefficients to use φ instead of ϕ .

MSBB Notation used by Moran, Shapiro, Boettner, and Bailey, *Fundamentals of Engineering Thermodynamics*, Eighth Edition. Wiley: Kendallville, 2014. Loads the non-default package options `EUFGHaN`, `intensive-lowercase`, and `delta`; removes the left parenthesis in partial derivatives and redefines `\IGM`, `\IG`, `\expansivitysymbol`, `\allcomponents`, `\allbut`, `\Ft`, `\Helmholtz@symbol`, `\@intensive`, `\@specific`, `\fmix`, `\phimix`, and `\pressure@symbol` to fit their somewhat ill-advised notation. Also resets `\partialmolar` to match their use.

Prausnitz Notation used by Prausnitz, Lichtenthaler, and de Azevedo, *Molecular Thermodynamics of Fluid-Phase Equilibria*, Third Edition, Pearson, 1998. Loads the non-default package option `intensive-lowercase` and redefines `\fmix`, `\phimix`, `\fpure`, `\hipure`, `\residual`, `\allcomponents`, and `\allbut` to fit their notation.

Sandler Notation used by Sandler, *Chemical, Biochemical, and Engineering Thermodynamics*, Fifth Edition. Wiley: Hoboken, 2017. Loads non-default package options `EUAGHaN` and `extensive-plain`; also redefines `\sat`, `\excess`, `\residual`, `\ncomponents`, `\fmix`, `\fstd`, `\phimix`, `\allcomponents`, `\IS`, `\Deltamix`, `\Deltarxn`, `\Deltasub`, `\Deltafus`, `\Deltavap`, and `\Henryrat`. Also redefines `\cV`, `\cP`, `\cVt`, and `\cPt`.

SVNAS Notation used by Smith, Van Ness, Abbott, and Swihart, *Introduction to Chemical Engineering Thermodynamics*, Ninth Edition. McGraw-Hill: Boston, 2021. Loads the non-default package option `extensive-superscript` and redefines `\allcomponents`, `\allbut`, `\IG`, `\IGM`, `\IS`, `\expansivitysymbol`, and `\Deltarxn` to fit their notation. The extensive heat capacities are also redefined, as they do not use such an entity.

TesterModell Notation used by Tester and Modell, *Thermodynamics and Its Applications*, Third Edition, Prentice Hall: Upper Saddle River, 1997. Loads the non-default package options `EUAGHaN` and `delta`; also redefines `\ncomponents` to be n and redefines `\allcomponents`, `\allbut`, and `\allbutlastand` to fit their (somewhat inconsistent) notation. Also redefines `\IG`, `\IGM`, `\IS`, `\excess`, `\reaction`, `\Henryrat`, `\Henrymol`, `\gammarat`, and `\gammamol` to fit their style.

Thompson Notation used by Thompson, *A Unified Introduction to Chemical Engineering Thermodynamics*, Stillwater Press: Orono, 2000. Loads the non-default package options `EUAGHaN` and `delta`; also redefines `\excess`, `\residual`, `\allcomponents`, `\ncomponents`, `\IS`, `\IG`, `\IGM`, `\fpure`, `\hipure`, `\mix`, `\Deltamix`, `\Deltarxn`, `\@intensive`, and `\allbut` to match his notation. Note that he uses c , n_C , and n for the number of components in various places in the book; I chose c for the definition of `\ncomponents`, but it is impossible to be completely consistent with his notation.

There may well be some inconsistencies between the notation in these books and the symbols used here. I will fix such inconsistencies as I become aware of them.

4 Implementation

This package requires the `amstext` package, as `\text` is used to handle `\sat`, `\IS`, `\IG`, `\IGM`, `\Henrymol`, `\fusion`, `\reaction`, `\vaporization`, and `\sublimation` by default, as well as several other macros defined by package options.

```
1 \RequirePackage{amstext}
```

4.1 Symbols Controlled by Package Options

We set some symbols prior to declaring the package options. The default symbols follow package option `EUAGHan`, even though the macros follow the option `EUFGHAN`.

`\dbar` The way `\dbar` is defined depends on the typeface you are using. We try to determine, at `\begin{document}`, which typeface you chose based on the packages that are loaded and some of their internal definitions. The `thermodynamics` package currently supports Computer Modern (the default or through `lmodern`), Palatino (through `pxfonts` or `newpxmath`), Times (through `txfonts`, `mathptmx`, or `newtxmath`), Utopia (through `mathdesign`), Bitstream Charter (through `mathdesign`), and Garamond (through `mathdesign`). Definitions of `\dbar` (with `\newcommand*` or `\providecommand*`) in the preamble will override the ones here.

```
2 \AtBeginDocument{
3   \@ifpackageloaded{pxfonts}{%
4     \providecommand*\dbar{{\mkern5mu\mathchar'26\mkern-10mu d}}%
5   }{}
6   \@ifpackageloaded{newpxmath}{%
7     \providecommand*\dbar{{\mkern5mu\mathchar'26\mkern-10mu d}}%
8   }{}
9   \@ifpackageloaded{txfonts}{%
10    \providecommand*\dbar{{\mkern5mu\mathchar'26\mkern-11mu d}}%
11  }{}
12  \@ifpackageloaded{mathptmx}{%
13    \providecommand*\dbar{{\mkern5mu\mathchar'26\mkern-10mu d}}%
14  }{}
15  \@ifpackageloaded{newtxmath}{%
16    \providecommand*\dbar{{\mkern5mu\mathchar'26\mkern-12mu d}}%
17  }{}
18  \@ifpackageloaded{mathdesign}{%
19    \def\@charter{mdbch}%
20    \def\@utopia{mdput}%
21    \def\@garamond{mdugm}%
22    \ifx\MD@default@family\@utopia
23      \providecommand*\dbar{{\mkern8mu\mathchar'26\mkern-20mu d}}%
24    \fi
25    \ifx\MD@default@family\@charter
26      \providecommand*\dbar{{\mkern5mu\mathchar'26\mkern-15mu d}}%
27    \fi
28    \ifx\MD@default@family\@garamond
29      \providecommand*\dbar{{\mkern5mu\mathchar'26\mkern-17mu d}}%
30    \fi
31  }{}
32  % Defaults to Computer Modern
33  \providecommand*\dbar{{\mkern3mu\mathchar'26\mkern-12mu d}}
```

Symbols are defined for the total energy, internal energy, Helmholtz free energy, Gibbs free energy, enthalpy, entropy, surface area, volume, number of moles, heat, and work; these are E , U , A , G , H , S , a , V , n , Q , and W , respectively. These commands should not be used on their own, but rather accessed through the macros `\Ut`, `\Um`, and `\Us` (using the internal energy as an example).

`\totalenergy@symbol` The default symbols are not intended to be easy to change—the intended mechanism
`\internalenergy@symbol` is through package options. If you want to use a non-standard symbol that is not
`\Helmholtz@symbol` available through one of the package options, you can redefine these macros inside
`\Gibbs@symbol` `\makeatletter... \makeatother`. For example,

```

\Landau@symbol      \makeatletter
\enthalpy@symbol    \renewcommand*\Helmholtz@symbol{H}
\entropy@symbol     \renewcommand*\enthalpy@symbol{h}
\area@symbol        \makeatother
\volume@symbol
\mole@symbol
\heat@symbol
\work@symbol
\temperature@symbol
\pressure@symbol

```

would define the ill-advised notation that I have nonetheless heard of that uses H for Helmholtz free energy and h for enthalpy. Note that the macros for temperature and pressure are only used inside the definitions of the compressibilities, expansivities, and heat capacities; there is no user-level macro for the temperature or the pressure, so it is up to the user to use consistent symbols for those properties.

```

35 \newcommand*\totalenergy@symbol{E}
36 \newcommand*\internalenergy@symbol{U}
37 \newcommand*\Helmholtz@symbol{A}
38 \newcommand*\Gibbs@symbol{G}
39 \newcommand*\Landau@symbol{\Omega}
40 \newcommand*\enthalpy@symbol{H}
41 \newcommand*\entropy@symbol{S}
42 \newcommand*\area@symbol{a}
43 \newcommand*\volume@symbol{V}
44 \newcommand*\mole@symbol{n}
45 \newcommand*\heat@symbol{Q}
46 \newcommand*\work@symbol{W}
47 \newcommand*\temperature@symbol{T}
48 \newcommand*\pressure@symbol{P}

```

`\thermo@underline` We then define two macros and several lengths that we shall use when drawing rules
`\thermo@overline` above or below a symbol. The default is to use underlined symbols for extensive quantities, plain symbols for molar quantities, and carets for specific quantities, but this can be changed using package options.

```

49 \newcommand*\thermo@underline[1]{%
50   \mkern1mu\underline{\mkern-1mu #1\mkern-4mu}\mkern4mu%
51 }
52 \newcommand*\thermo@overline[1]{%
53   \mkern2mu\overline{\mkern-2mu #1\mkern-1mu}\mkern1mu%
54 }

```

`\PartialOpen` We define three commands to use to denote the beginning and end of partial deriva-
`\PartialClose` tives. These symbols can be customized by package options. Default is parentheses,
`\PartialEmptyClose`

meaning that `\[\Partial{f}{x}{y} \]` renders as

$$\left(\frac{\partial f}{\partial x}\right)_y$$

with the defaults. The macro `\PartialEmptyClose` is used when the last argument to `\Partial` is empty, which is important for the bar option to the document class or inside the thermobar environment.

```
55 \newcommand*\PartialOpen{({}
56 \newcommand*\PartialClose{)}}
57 \newcommand*\PartialEmptyClose{)}
```

4.2 Package Options

We declare a bunch of options for which sets of symbols to use. These are summarized in Table 3.

```
58 \DeclareOption{EUAGHan}{}% the default
59 \DeclareOption{EUAGHaN}{\renewcommand*\mole@symbol{N}}%
60 \DeclareOption{EUHAGan}{\ExecuteOptions{EUAGHan}}
61 \DeclareOption{EUHAGaN}{\ExecuteOptions{EUAGHaN}}
62 \DeclareOption{EUFGHan}{% this is my favorite set of symbols
63  \renewcommand*\Helmholtz@symbol{F}%
64  \renewcommand*\area@symbol{A}%
65 }
66 \DeclareOption{EUFGHaN}{% this is my second favorite set of symbols
67  \renewcommand*\Helmholtz@symbol{F}%
68  \renewcommand*\area@symbol{A}%
69  \renewcommand*\mole@symbol{N}%
70 }
71 \DeclareOption{EEFGHan}{% this treats all energy the same way(?)
72  \renewcommand*\internalenergy@symbol{E}%
73  \renewcommand*\Helmholtz@symbol{F}%
74  \renewcommand*\area@symbol{A}%
75 }
76 \DeclareOption{EEFGHaN}{% this treats all energy the same way(?)
77  \renewcommand*\internalenergy@symbol{E}%
78  \renewcommand*\Helmholtz@symbol{F}%
79  \renewcommand*\area@symbol{A}%
80  \renewcommand*\mole@symbol{N}%
81 }
82 \DeclareOption{EEFGHan}{%
83  \renewcommand*\internalenergy@symbol{E}%
84  \renewcommand*\Helmholtz@symbol{F}
85 }
86 \DeclareOption{EEFGHaN}{%
87  \renewcommand*\internalenergy@symbol{E}%
88  \renewcommand*\Helmholtz@symbol{F}
89  \renewcommand*\mole@symbol{N}%
90 }
91 \DeclareOption{EEAGHan}{%
92  \renewcommand*\internalenergy@symbol{E}%
93 }
94 \DeclareOption{EEAGHaN}{%
```

```

95 \renewcommand*{\internalenergy@symbol}{E}%
96 \renewcommand*{\mole@symbol}{N}%
97 }
98 \DeclareOption{EUAGHAN}{% Helmholtz free energy and area look the same here
99 \renewcommand*{\area@symbol}{A}%
100 }
101 \DeclareOption{EUAGHAN}{% Helmholtz free energy and area look the same here
102 \renewcommand*{\area@symbol}{A}%
103 \renewcommand*{\mole@symbol}{N}%
104 }
105 \DeclareOption{EUFGHan}{%
106 \renewcommand*{\Helmholtz@symbol}{F}%
107 }
108 \DeclareOption{EUFGHan}{%
109 \renewcommand*{\Helmholtz@symbol}{F}%
110 \renewcommand*{\mole@symbol}{N}%
111 }

```

The delta option redefines `\dbar` to produce the symbol δ . The default is to use a d with a slash through it (\bar{d}) for inexact differentials unless the user overrides it with this option. The macro could also be redefined manually, of course.

```

112 \DeclareOption{delta}{\let\dbar\delta}

```

`\@extensive` Next, we define options for the set of notation. The default is `intensive-plain`,
`\@intensive` which produces things like V for molar volume, \underline{V} for total volume, and \hat{V} for specific
`\@specific` volume. These are defined via the internal macros `\@extensive`, `\@intensive`, and
`\@specific`.

```

113 \let\@extensive\relax
114 \let\@intensive\relax
115 \let\@specific\hat

116 \DeclareOption{extensive-plain}{%
117 \let\@extensive\relax
118 \let\@intensive\thermo@underline
119 }
120 \DeclareOption{intensive-plain}{% the default
121 \let\@extensive\thermo@underline
122 \let\@intensive\relax
123 }
124 \DeclareOption{intensive-lowercase}{% PLEASE don't use this!
125 \let\@extensive\MakeUppercase
126 \let\@intensive\MakeLowercase
127 \def\@specific#1{\MakeLowercase{\hat #1}}
128 \AtEndOfPackage{
129 \renewcommand*{\partialmolar}[1]{%
130 \gdef\pm@symbol{\MakeLowercase #1}\generic@pm}
131 \renewcommand*{\heatcapacitysymbol}{c}
132 }
133 }
134 \DeclareOption{extensive-superscript}{%
135 \gdef\@extensive#1{\#1^t}%
136 \let\@intensive\relax%
137 \AtEndOfPackage{%
138 \renewcommand*{\URt}{\internalenergy@symbol^{\residual,t}}

```

```

139     \renewcommand*{\HRt}{\enthalpy@symbol^\residual,t}}
140     \renewcommand*{\FRt}{\Helmholtz@symbol^\residual,t}}
141     \renewcommand*{\GRt}{\Gibbs@symbol^\residual,t}}
142     \renewcommand*{\VRt}{\volume@symbol^\residual,t}}
143     \renewcommand*{\SRt}{\entropy@symbol^\residual,t}}
144     \renewcommand*{\Uet}{\internalenergy@symbol^\excess,t}}
145     \renewcommand*{\HET}{\enthalpy@symbol^\excess,t}}
146     \renewcommand*{\FET}{\Helmholtz@symbol^\excess,t}}
147     \renewcommand*{\GET}{\Gibbs@symbol^\excess,t}}
148     \renewcommand*{\VET}{\volume@symbol^\excess,t}}
149     \renewcommand*{\SET}{\entropy@symbol^\excess,t}}
150   }
151 }

```

The next two options choose whether variables held constant are subscripted (the default) or placed next to the function. The `nosubscripts` option currently requires that the variable being changed is the first one in the argument list. The difference is

$$\left(\frac{\partial U}{\partial S}\right)_V \quad \text{versus} \quad \left(\frac{\partial U(S, V)}{\partial S}\right)$$

for `subscripts` and `nosubscripts`, respectively.

```

152 \newif\if@subscripted
153 \@subscriptedtrue
154 \DeclareOption{subscripts}{\@subscriptedtrue}
155 \DeclareOption{nosubscripts}{\@subscriptedfalse}

```

These options change how `\Partial` and friends render derivatives. The default is parentheses, but other options include brackets, braces, a vertical bar on the right side, or plain (undecorated) derivatives.

```

156 \DeclareOption{parentheses}{}
157 \DeclareOption{brackets}{%
158   \renewcommand*{\PartialOpen}{[}%
159   \renewcommand*{\PartialClose}{]}%
160   \renewcommand*{\PartialEmptyClose}{]}%
161 }
162 \DeclareOption{braces}{%
163   \renewcommand*{\PartialOpen}{\{}%
164   \renewcommand*{\PartialClose}{\}}%
165   \renewcommand*{\PartialEmptyClose}{\}}%
166 }
167 \DeclareOption{bar}{%
168   \renewcommand*{\PartialOpen}{.}%
169   \renewcommand*{\PartialClose}{\rvert}%
170   \renewcommand*{\PartialEmptyClose}{.}%
171 }
172 \DeclareOption{plain-derivatives}{% This implies dU(S,V,N)/dS notation
173   \renewcommand*{\PartialOpen}{.%
174   \renewcommand*{\PartialClose}{.%
175   \renewcommand*{\PartialEmptyClose}{.%
176   \ExecuteOptions{nosubscripts}%
177 }

```

4.3 The Number of Moles Macros

`\ncomponents` We define the number of components, default C , for use in the “all moles” and related macros.

```
178 \newcommand*\ncomponents{C}
```

`\allNs` Several macros define a shorthand for “moles of all species” (`\allNs`) and “moles of all species except” (`\allNsbut`), as well as similar quantities for masses (`\allMs`, `\allMsbut`) and chemical potentials (`\allmus`, `\allmusbut`), which occur frequently in mixture thermodynamics. The default is for `\allNs` to become \vec{n} and `\allNsbut{i}` to become $n_{j\neq i}$. The optional argument changes which index (default: j) to use in the left side of the inequality.² Essentially identical commands are defined for chemical potentials and masses: `\allmus` and `\allmusbut` and `\allMs` and `\allMsbut`, respectively.

```
179 \newcommand*\allNs{\allcomponents{Nt}}
```

```
180 \newcommand*\allXs{\allcomponents{x}}
```

```
181 \newcommand*\allYs{\allcomponents{y}}
```

```
182 \newcommand*\allmus{\allcomponents{\mu}}
```

```
183 \newcommand*\allMs{\allcomponents{m}}
```

```
184 \newcommand*\allWs{\allcomponents{w}}%
```

`\allXs` Similar commands are defined for mole fractions (`\allXs`, `\allYs`, etc.), but these `\allYs` assume the last mole fraction is *not* one of the variables—that is, `\allXsbut` and `\allWs` `\allYsbut` assume the argument *and* `\ncomponents` are held constant. For example,

```
\allWsbut    \[ \Partial{\Gm}{T}{P,\allXs} = -\Sm \quadquad
\allXsbut    \Partial{\Gm}{x_i}{T,P,\allXsbut{i}} \neq \Gm_i \]
\allYsbut
```

yields

$$\left(\frac{\partial G}{\partial T}\right)_{P,\vec{x}} = -S \quad \left(\frac{\partial G}{\partial x_i}\right)_{T,P,x_{j\neq i},C} \neq \bar{G}_i.$$

```
185 \newcommand*\allNsbut}[2][j]{\allbut[#1]{#2}{Nt}}
```

```
186 \newcommand*\allXsbut}[2][j]{\allbutlastand[#1]{#2}{x}}
```

```
187 \newcommand*\allYsbut}[2][j]{\allbutlastand[#1]{#2}{y}}
```

```
188 \newcommand*\allmusbut}[2][j]{\allbut[#1]{#2}{\mu}}
```

```
189 \newcommand*\allMsbut}[2][j]{\allbut[#1]{#2}{m}}
```

```
190 \newcommand*\allWsbut}[2][j]{\allbutlastand[#1]{#2}{w}}%
```

`\allbutlastand` The `\allcomponents`, `\allbut`, and `\allbutlastand` macros can be used to define new entities; say, if you want to use z_i as a mole fraction, then use

```
\allbut
\allcomponents \newcommand*\allZsbut}[2][j]{\allbutlastand[#1]{#2}{z}}
```

Similarly, something meaning the concentrations of every species could be defined via

```
\newcommand*\allCs{\allcomponents{C}}
```

```
191 \newcommand*\allcomponents}[1]{\vec{#1}}
```

```
192 \newcommand*\allbut}[3][j]{%
```

```
193 \def\tmp@arg{#2}%
```

```
194 \def\tmp@@arg{#1}%
```

²The index j is automatically replaced with k if the user issues `\allNsbut{j}`.

```

195 \ifx\tmp@arg\tmp@@arg
196   {#3}_{k\neq #2}%
197 \else
198   {#3}_{#1\neq #2}%
199 \fi
200 }
201 \newcommand*\allbutlastand}[3][j]{%
202   \edef\tmp@arg{#1}%
203   \edef\tmp@@arg{#2}%
204   \ifx\tmp@@arg\ncomponents
205     {#3}_{#1 \neq #2}%
206   \else
207     \ifx\tmp@arg\tmp@@arg
208       {#3}_{k \neq #2,\ncomponents}%
209     \else
210       {#3}_{#1 \neq #2,\ncomponents}%
211     \fi
212   \fi
213 }

```

We then define two package options that change how to render \allNs and friends.

```

214 \DeclareOption{moles-index}{}
215 \DeclareOption{moles-range}{%
216   \renewcommand*\allcomponents}[1]{{#1}_1,\dots,{#1}_{\ncomponents}}
217   \renewcommand*\allbut}[3][j]{%
218     \def\@one{1}%
219     \edef\tmp@arg{#2}%
220     \ifx\tmp@arg\@one
221       {#3}_2,\dots,{#3}_{\ncomponents}%
222     \else
223       \ifx\tmp@arg\ncomponents
224         {#3}_1,\dots,{#3}_{\ncomponents-1}%
225       \else
226         {#3}_1,\dots,[[#3]_{#2}],\dots,{#3}_{\ncomponents}%
227       \fi
228     \fi
229   }
230   \renewcommand*\allbutlastand}[3][j]{%
231     \def\@one{1}%
232     \edef\tmp@arg{#2}%
233     \edef\@ncminusone{\ncomponents-1}%
234     \ifx\tmp@arg\@one
235       {#3}_2,\dots,{#3}_{\ncomponents-1}%
236     \else
237       \ifx\tmp@arg\@ncminusone
238         {#3}_1,\dots,{#3}_{\ncomponents-2}%
239       \else
240         \ifx\tmp@arg\ncomponents
241           {#3}_1,\dots,{#3}_{\ncomponents-1}%
242         \else
243           {#3}_1,\dots,[[#3]_{#2}],\dots,{#3}_{\ncomponents}%
244         \fi
245       \fi
246     \fi

```



```

247 }
248 }

```

The remaining options define textbook-specific notation.

```

249 \DeclareOption{Bejan}{
250   \ExecuteOptions{EUFGHAN,intensive-lowercase,delta}
251   \let\@specific\MakeLowercase
252   \def\@intensive#1{\MakeLowercase{\bar #1}}
253   \AtEndOfPackage{
254     \DeclareSubscrSymbol{cV}{\bar\heatcapacitysymbol}{v}
255     \DeclareSubscrSymbol{cVs}{\heatcapacitysymbol}{v}
256     \DeclareSubscrSymbol{cVt}{\Nt\heatcapacitysymbol}{v}
257     \renewcommand*\expansivitiysymbol{\beta}
258     \renewcommand*\ncomponents{n}
259   }
260 }
261 \DeclareOption{CBK}{
262   \ExecuteOptions{EUAGHAN,intensive-lowercase}
263   \AtEndOfPackage{
264     \let\thermo@overline\widetilde
265     \let\@specific\MakeLowercase
266     \renewcommand*\@intensive[1]{\MakeLowercase{\bar{#1}}}
267     \renewcommand*\pressure@symbol{p}
268     \renewcommand*\Deltarxn[1]{#1_R}
269     \renewcommand*\compressibilitysymbol{\beta}
270     \renewcommand*\expansivitiysymbol{\alpha}
271   }
272 }
273 \DeclareOption{ElliottLira}{
274   \AtEndOfPackage{
275     \renewcommand*\allcomponents[1]{#1}
276     \renewcommand*\Deltarxn[1]{\Delta #1}
277   }
278 }
279 \DeclareOption{Koretsky}{
280   \ExecuteOptions{EUAGHAN,brackets,intensive-lowercase,delta}
281   \AtEndOfPackage{
282     \renewcommand*\partialmolar[1]{\gdef\pm@symbol{#1}\generic@pm}
283     \renewcommand*\expansivitiysymbol{\beta}
284     \renewcommand*\IS{{\text{ideal}}}
285     \renewcommand*\residual{{\text{dep}}}
286     \renewcommand*\IG{{\text{ideal}}}
287     \renewcommand*\IGM{{\text{ideal}}}
288     \renewcommand*\Deltafus[1]{\Delta{#1}_\fusion}
289     \renewcommand*\Deltasub[1]{\Delta{#1}_\sublimation}
290     \renewcommand*\Deltavap[1]{\Delta{#1}_\vaporization}
291     \renewcommand*\Henryrat{{\mathcal{H}}}
292     \renewcommand*\gammarat{\gamma^\text{Henry's}}
293     \let\hipure\varphi
294     \renewcommand*\phimix{\hat\varphi}
295     \renewcommand*\phisat{\varphi^\text{sat}}
296     \let\Delta@fus@sym\relax
297     \DeclareSubscrSymbol{@Deltafus}{\Delta@fus@sym}{\fusion}
298     \renewcommand*\Deltafus[1]{\def\Delta@fus@sym{\Delta #1}@Deltafus}

```

```

299 \let\Delta@vap@sym\relax
300 \DeclareSubscrSymbol{@Deltavap}{\Delta@vap@sym}{\vaporization}
301 \renewcommand*{\Deltasub}[1]{\def\Delta@sub@sym{\Delta #1}\@Deltasub}
302 \let\Delta@sub@sym\relax
303 \DeclareSubscrSymbol{@Deltasub}{\Delta@sub@sym}{\sublimation}
304 \renewcommand*{\Deltasub}[1]{\def\Delta@sub@sym{\Delta #1}\@Deltasub}
305 }
306 }
307 \DeclareOption{MSBB}{
308 \ExecuteOptions{EUFGHAn,intensive-lowercase,delta}
309 \AtEndOfPackage{
310 \renewcommand*{\IGM}{\ast}
311 \renewcommand*{\IG}{\ast}
312 \renewcommand*{\expansivitysymbol}{\beta}
313 \renewcommand*{\allcomponents}[1]{#1}
314 \renewcommand*{\allbut}[3][j]{%
315 \def\tmp@arg{#2}%
316 \def\tmp@@arg{#1}%
317 \ifx\tmp@arg\tmp@@arg
318 {#3}_{k}%
319 \else
320 {#3}_{#1}%
321 \fi
322 }
323 \let\Helmholtz@symbol\psi
324 \renewcommand*{\Ft}{\Psi}
325 \def\@intensive#1{\MakeLowercase{\thermo@overline #1}}
326 \let\@specific\MakeLowercase
327 \renewcommand*{\partialmolar}[1]{\gdef\pm@symbol{#1}\generic@pm}
328 \renewcommand*{\fmix}{\bar f}
329 \renewcommand*{\phimix}{\bar\phi}
330 \renewcommand*{\pressure@symbol}{p}
331 }
332 }
333 \DeclareOption{Prausnitz}{
334 \ExecuteOptions{intensive-lowercase}
335 \AtEndOfPackage{
336 \renewcommand*{\fmix}{f}
337 \let\phimix\phi
338 \renewcommand*{\fsat}{\fpure^{\sat}}
339 \DeclareSubscrSymbol{fpure}{f}{\text{pure}}
340 \DeclareSubscrSymbol{phipure}{\phi}{\text{pure}}
341 \renewcommand*{\residual}{\mathcal{R}}
342 \renewcommand*{\allcomponents}[1]{#1_i}
343 \renewcommand*{\allbut}[3][j]{%
344 \def\tmp@arg{#2}%
345 \def\tmp@@arg{#1}%
346 \ifx\tmp@arg\tmp@@arg
347 {#3}_{k}%
348 \else
349 {#3}_{#1}%
350 \fi
351 }
352 }

```

```

353 }
354 \DeclareOption{Sandler}{
355   \ExecuteOptions{EUAGHaN,extensive-plain}
356   \AtEndOfPackage{
357     \renewcommand*\sat{{\text{vap}}}
358     \renewcommand*\excess{{\text{ex}}}
359     \renewcommand*\residual{{\text{r}}}
360     \renewcommand*\ncomponents{\mathcal{C}}
361     \renewcommand*\fmix{\bar f}
362     \renewcommand*\fstd{\bar f^\circ}
363     \renewcommand*\phimix{\bar\phi}
364     \renewcommand*\allcomponents[1]{\thermo@underline{#1}}
365     \renewcommand*\IG{{\text{IG}}}
366     \renewcommand*\IGM{{\text{IGM}}}
367     \renewcommand*\IS{{\text{IM}}}
368     \renewcommand*\Deltamix[1]{\Delta_\mixing #1}
369     \renewcommand*\Deltarxn[1]{\Delta_\reaction #1}
370     \renewcommand*\Deltasub[1]{\Delta_\sublimation #1}
371     \renewcommand*\Deltafus[1]{\Delta_\fusion #1}
372     \renewcommand*\Deltavap[1]{\Delta_\vaporization #1}
373     \renewcommand*\Henryrat}{H}
374     \DeclareSubscrSymbol{cV}{\heatcapacitiesymbol}{\volume@symbol}
375     \DeclareSubscrSymbol{cP}{\heatcapacitiesymbol}{\pressure@symbol}
376     \DeclareSubscrSymbol{cVt}{\Nt\heatcapacitiesymbol}{\volume@symbol}
377     \DeclareSubscrSymbol{cPt}{\Nt\heatcapacitiesymbol}{\pressure@symbol}
378   }
379 }
380 \DeclareOption{SVNAS}{
381   \ExecuteOptions{extensive-superscript}
382   \AtEndOfPackage{
383     \renewcommand*\allcomponents[1]{#1}
384     \renewcommand*\allbut[3][j]{%
385       \def\tmp@arg{#2}%
386       \def\tmp@@arg{#1}%
387       \ifx\tmp@arg\tmp@@arg
388         {#3}_{k}%
389       \else
390         {#3}_{#1}%
391       \fi
392     }
393     \renewcommand*\IG{{ig}}
394     \renewcommand*\IGM{{ig}}
395     \renewcommand*\IS{{id}}
396     \renewcommand*\expansivitesymbol{\beta}
397     \renewcommand*\Deltarxn[1]{\Delta #1}
398     \DeclareSubscrSymbol{cVt}{\Nt\heatcapacitiesymbol}{\volume@symbol}
399     \DeclareSubscrSymbol{cPt}{\Nt\heatcapacitiesymbol}{\pressure@symbol}
400   }
401 }
402 \DeclareOption{TesterModell}{
403   \ExecuteOptions{EUAGHaN,delta}
404   \AtEndOfPackage{
405     \renewcommand*\ncomponents}{n}
406     \renewcommand*\allcomponents[1]{{#1}_i}

```

```

407 \renewcommand*{\allbut}[3][j]{%
408 \def\tmp@arg{#2}%
409 \def\tmp@@arg{#1}%
410 \ifx\tmp@arg\tmp@@arg
411 {#3}_k\relax[#2]
412 \else
413 {#3}_{#1}\relax[#2]
414 \fi
415 }
416 \renewcommand*{\allbutlastand}[3][j]{%
417 \edef\tmp@arg{#1}%
418 \edef\tmp@@arg{#2}%
419 \ifx\tmp@@arg\ncomponents
420 {#3}_{#1}\relax[#2]%
421 \else
422 \ifx\tmp@arg\tmp@@arg
423 {#3}\relax[#2,\ncomponents]%
424 \else
425 {#3}\relax[#2,\ncomponents]%
426 \fi
427 \fi
428 }
429 \renewcommand*{\IG}{{ig}}
430 % Tester & Modell never use "igm" anywhere, and I only found ig once
431 \renewcommand*{\IGM}{{igm}}
432 \renewcommand*{\IS}{{ID}}
433 \renewcommand*{\excess}{{EX}}%
434 \renewcommand*{\reaction}{{rx}}
435 \renewcommand*{\Henryrat}{{f^{\ast\ast}}}
436 \renewcommand*{\Henrymol}{{f^{\ast}}}
437 \renewcommand*{\gammarat}{{\gamma^{\ast\ast}}}
438 \renewcommand*{\gammamol}{{\gamma^{\ast}}}
439 }
440 }
441 \DeclareOption{Thompson}{
442 \ExecuteOptions{EUAGHAn,delta}
443 \AtEndOfPackage{
444 \renewcommand*{\excess}{{EX}}
445 \renewcommand*{\residual}{{R}}
446 \renewcommand*{\allcomponents}[1]{{#1}_j}
447 \renewcommand*{\ncomponents}{c}
448 \renewcommand*{\IS}{{IS}}
449 \renewcommand*{\IG}{{IG}}
450 \let\IGM\IG
451 \def\@fpure_#1{f_{#1}\@ifnextchar^{{}\bullet}}
452 \renewcommand*{\fpure}{{\@ifnextchar_{\@fpure}{f}}
453 \def\@phi@pure_#1{\phi_{#1}\@ifnextchar^{{}\bullet}}
454 \renewcommand*{\phipure}{{\@ifnextchar_{\@phi@pure}{\phi}}
455 \renewcommand*{\mixing}{{MIX}}
456 \renewcommand*{\Deltamix}[1]{\Delta_\mixing #1}
457 \renewcommand*{\Deltarxn}[1]{\Delta #1}
458 \def\@@intensive_#1_{#1}\@ifnextchar^{{}\bullet}}
459 \newcommand*{\@intensive}[1]{#1\@ifnextchar_{\@@intensive}}
460 \renewcommand*{\allbut}[3][j]{%

```

```

461 \def\tmp@arg{#2}%
462 \def\tmp@@arg{#1}%
463 \ifx\tmp@arg\tmp@@arg
464   {#3}_{k}\neq{#3}_{#2}%
465 \else
466   {#3}_{#1}\neq{#3}_{#2}%
467 \fi
468 }
469 }
470 }

```

We execute the default options below.

```

471 \ExecuteOptions{EUAGHan,subscripts,parentheses,intensive-plain,moles-index}
472 \ProcessOptions

```

4.4 Commands for Partial Derivatives

The `\Partial` command and its second-order siblings are defined as below. They typeset partial derivatives of the first argument with respect to the second (and third, in the case of mixed second partial derivatives) arguments, holding the last argument constant.

The starred forms adjust the spacing after the partial derivative so the trailing binary operator (assumed to be the same width as an equals sign) overhangs the variables held constant. We thus set `operator@width` to be *just* greater than the width of an equals sign.

```

473 \newlength{\Partial@const@width}
474 \newlength{\operator@width}
475 \settowidth{\operator@width}{=}
476 \newlength{\adjust@width}
477 \setlength{\adjust@width}{0.1\operator@width}
478 \addtolength{\operator@width}{\adjust@width}

```

`\Partial*` The command `\Partial` and its friends drastically simplify the creation of partial derivatives. The command `\Partial*` is the same as `\Partial` except that it adjusts the spacing so the (presumably) binary operator that follows it slightly overlaps the subscripts.

```

479 \def\Partial@start{\left\PartialOpen}
480 \def\Partial@end{\right\PartialClose}
481 \def\Partial@empty@end{\right\PartialEmptyClose}
482 \def\Partial{\@ifstar\Partial@star\Partial@nostar}
483 %^^A Several ifs to use later (breaks it if these definitions are inside)
484 \newif\iftwo@has@Nt
485 \two@has@Ntfalse
486 \newif\iftwo@has@xory
487 \two@has@xoryfalse
488 \newif\ifthree@has@Nt
489 \three@has@Ntfalse
490 \newif\ifthree@has@xory
491 \three@has@xoryfalse
492 \AtEndOfPackage{%
493 \edef\thermo@Nt{\Nt}%
494 \edef\thermo@x{x}%

```

```

495 \edef\thermo@y{y}%
496 }
497 \def\thermo@parse@two#1\relax{%
498 \@tfor\@i:=#1\do{%
499 \edef\@@i{\@i}%
500 \ifx\@@i\thermo@Nt
501 \two@has@Nttrue
502 \@break@tfor
503 \fi
504 \ifx\@@i\thermo@x
505 \two@has@xorytrue
506 \@break@tfor
507 \fi
508 \ifx\@@i\thermo@y
509 \two@has@xorytrue
510 \@break@tfor
511 \fi
512 }
513 }
514 \newcommand*\Partial@nostar}[3]{%
515 \def\tmp@arg{#3}%
516 \if@subscripted
517 % Handles situation of empty variables held constant
518 \ifx\tmp@arg@empty
519 \ensuremath{\Partial@start\frac{\partial #1}
520 {\partial #2}\Partial@empty@end}%
521 \else
522 \ensuremath{\Partial@start\frac{\partial #1}
523 {\partial #2}\Partial@end_{#3}}%
524 \fi
525 \else
526 % If not subscripted, we put the arguments (if any) in the derivative
527 \ifx\tmp@arg@empty
528 \ensuremath{\Partial@start\frac{\partial #1}
529 {\partial #2}\Partial@empty@end}%
530 \else
531 % Check for whether #3 contains \allNsbut{i}/etc. and #2 is \Nt_i/etc.
532 \thermo@parse@two#2\relax
533 \iftwo@has@Nt
534 \renewcommand*\allbut}[3][j]{\allcomponents{##3}}%
535 \ensuremath{\Partial@start\frac{\partial #1(#3)}
536 {\partial #2}\Partial@end}%
537 \else
538 \iftwo@has@xory
539 \renewcommand*\allbutlastand}[3][j]{\allbut{\ncomponents}{##3}}%
540 \ensuremath{\Partial@start\frac{\partial #1(#3)}
541 {\partial #2}\Partial@end}%
542 \else
543 \ensuremath{\Partial@start\frac{\partial #1(#2,#3)}
544 {\partial #2}\Partial@end}%
545 \fi
546 \fi
547 \fi
548 \fi

```

```

549 }
550 \newcommand*{\Partial@star}[3]{%
551   \settowidth{\Partial@const@width}{\ensuremath{#3}}%
552   \addtolength{\Partial@const@width}{-0.15\Partial@const@width}%
553   \Partial@nostar{#1}{#2}{#3}%
554   \if@subscripted
555     \ifdim\operator@width<\Partial@const@width
556       \kern -\operator@width
557     \else
558       \kern -\Partial@const@width
559     \fi
560   \fi
561 }

```

`\PartialBigg` The `\PartialBigg` macro (and its starred form) replace the `\left` and `\right` commands in `\Partial` with amsmath's `\Biggl` and `\Biggr` variants. The starred form is inherited from `\Partial` without modification.

```

562 \def\PartialBigg{%
563   \def\Partial@start{\Biggl\PartialOpen}%
564   \def\Partial@end{\Biggr\PartialClose}%
565   \def\Partial@empty@end{\Biggr\PartialClose}%
566   \Partial%
567 }

```

`\Partialbigg` The `\Partialbigg` macro does the same thing as `\PartialBigg`, except using amsmath's `\biggl`/`\biggr` variants.

```

568 \def\Partialbigg{%
569   \def\Partial@start{\biggl\PartialOpen}%
570   \def\Partial@end{\biggr\PartialClose}%
571   \def\Partial@empty@end{\biggr\PartialClose}%
572   \Partial%
573 }

```

`\PartialSecond` The second partial derivatives are defined similarly to `\Partial`.

```

\PartialSecond* 574 \def\PartialSecond{\@ifstar\PartialSecond@star\PartialSecond@nostar}
575 \newcommand*{\PartialSecond@nostar}[3]{%
576   \def\tmp@arg{#3}%
577   \if@subscripted
578     % Handles situation of empty variables held constant
579     \ifx\tmp@arg@empty
580       \ensuremath{\Partial@start\frac{\partial^2 #1}
581         {\partial #2^2}\Partial@empty@end}%
582     \else
583       \ensuremath{\Partial@start\frac{\partial^2 #1}
584         {\partial #2^2}\Partial@end_{#3}}%
585     \fi
586   \else
587     \ifx\tmp@arg@empty
588       \ensuremath{\Partial@start\frac{\partial^2 #1}
589         {\partial #2^2}\Partial@empty@end}%
590     \else
591       % Check for whether #3 contains \allNsbut{i}/etc. and #2 is \Nt_i/etc.
592       \thermo@parse@two#2\relax

```

```

593 \iftwo@has@Nt
594 \renewcommand*{\allbut}[3][j]{\allcomponents{##3}}%
595 \ensuremath{\Partial@start\frac{\partial^2 #1(#3)}
596 \partial #2^2}\Partial@end}%
597 \else
598 \iftwo@has@xory
599 \renewcommand*{\allbutlastand}[3][j]{\allbut{\ncomponents}{##3}}%
600 \ensuremath{\Partial@start\frac{\partial^2 #1(#3)}
601 \partial #2^2}\Partial@end}%
602 \else
603 \ensuremath{\Partial@start\frac{\partial^2 #1(#2,#3)}
604 \partial {#2}^2}\Partial@end}%
605 \fi
606 \fi
607 \fi
608 \fi
609 }
610 \newcommand*{\PartialSecond@star}[3]{%
611 \settowidth{\Partial@const@width}{\ensuremath{#3}}%
612 \addtolength{\Partial@const@width}{-0.15\Partial@const@width}%
613 \PartialSecond@nostar{#1}{#2}{#3}%
614 \if@subscripted
615 \ifdim\operator@width<\Partial@const@width
616 \kern -\operator@width
617 \else
618 \kern -\Partial@const@width
619 \fi
620 \fi
621 }

```

`\PartialSecondBigg` The `\PartialSecondBigg` macro and its starred variant replace `\left` and `\right` with `amsmath's \Bigl` and `\Biggr`.

```

622 \def\PartialSecondBigg{%
623 \def\Partial@start{\expandafter\Bigl\PartialOpen}%
624 \def\Partial@end{\expandafter\Biggr\PartialClose}%
625 \PartialSecond%
626 }

```

`\PartialSecondbigg` The `\PartialSecondbigg` macro and its starred variant replace `\left` and `\right` with `amsmath's \bigl` and `\biggr`.

```

627 \def\PartialSecondbigg{%
628 \def\Partial@start{\expandafter\bigl\PartialOpen}%
629 \def\Partial@end{\expandafter\biggr\PartialClose}%
630 \PartialSecond%
631 }

```

`\PartialMixSecond` The macro `\PartialMixSecond` takes an extra argument, but is otherwise the same as its same-variable cousin.

```

632 \def\PartialMixSecond{\@ifstar\PartialMixSecond@star\PartialMixSecond@nostar}
633 \newcommand*{\PartialMixSecond@nostar}[4]{%
634 \def\tmp@arg{#4}%
635 \if@subscripted
636 % Handles situation of empty variables held constant

```



```

637 \ifx\tmp@arg\@empty
638   \ensuremath{\Partial@start\frac{\partial^2 #1}
639             {\partial #2\partial#3}\Partial@end}%
640 \else
641   \ensuremath{\Partial@start\frac{\partial^2 #1}
642             {\partial #2\partial#3}\Partial@end_{#4}}%
643 \fi
644 \else
645   \ifx\tmp@arg\@empty
646     \ensuremath{\Partial@start\frac{\partial^2 #1(#2,#3)}
647               {\partial #2\partial#3}\Partial@end}%
648   \else
649     % This time, we check whether #2 OR #3 are \Nt_i/etc.
650     \thermo@parse@two#3\relax
651     \iftwo@has@Nt\three@has@Nttrue\fi
652     \iftwo@has@xory\three@has@xorytrue\fi
653     \two@has@Ntfalse
654     \two@has@xoryfalse
655     \thermo@parse@two#2\relax
656     \iftwo@has@Nt
657       \renewcommand*\allbut}[3][j]{\allcomponents{##3}}%
658       \ensuremath{\Partial@start\frac{\partial^2 #1(#3,#4)}
659                 {\partial #2\partial#3}\Partial@end}%
660     \else
661       \iftwo@has@xory
662         \renewcommand*\allbutlastand}[3][j]{\allbut{\ncomponents}{##3}}%
663         \ensuremath{\Partial@start\frac{\partial^2 #1(#3,#4)}
664                   {\partial #2\partial#3}\Partial@end}%
665       \else
666         \ifthree@has@Nt
667           \renewcommand*\allbut}[3][j]{\allcomponents{##3}}%
668           \ensuremath{\Partial@start\frac{\partial^2 #1(#2,#4)}
669                     {\partial #2\partial#3}\Partial@end}%
670         \else
671           \ifthree@has@xory
672             \PackageWarning{thermodynamics}{GOT HERE}
673             %\renewcommand*\allbutlastand}[3][j]{\allbut{\ncomponents}{##3}}%
674             \renewcommand*\allbutlastand}[3][j]{SCREW YOU}%
675             \ensuremath{\Partial@start\frac{\partial^2 #1(#2,#4)}
676                       {\partial #2\partial#3}\Partial@end}%
677           \else
678             \ensuremath{\Partial@start\frac{\partial^2 #1(#2,#3,#4)}
679                       {\partial #2\partial#3}\Partial@end}%
680           \fi
681         \fi
682       \fi
683     \fi
684   \fi
685 \fi
686 }
687 \newcommand*\PartialMixSecond@star}[4]{%
688   \settowidth{\Partial@const@width}{\ensuremath{#4}}%
689   \addtolength{\Partial@const@width}{-0.25\Partial@const@width}%
690   \PartialMixSecond@nostar{#1}{#2}{#3}{#4}%

```

```

691 \if@subscripted
692 \ifdim\operator@width<\Partial@const@width
693 \kern -\operator@width
694 \else
695 \kern -\Partial@const@width
696 \fi
697 \fi
698 }

```

`\PartialMixSecondBigg` The macro `\PartialMixSecondBigg` and its starred form are analogous to `\PartialBigg` and `\PartialSecondBigg`. `\PartialMixSecondbigg` and its starred form are similarly analogous to `\Partialbigg` and `\PartialSecondbigg`.

```

\PartialMixSecondbigg*
\PartialMixSecondbigg*
\PartialMixSecondbigg*
699 \def\PartialMixSecondBigg{%
700 \def\Partial@start{\expandafter\Biggl\PartialOpen}%
701 \def\Partial@end{\expandafter\Biggr\PartialClose}%
702 \PartialMixSecond%
703 }
704 \def\PartialMixSecondbigg{%
705 \def\Partial@start{\expandafter\biggl\PartialOpen}%
706 \def\Partial@end{\expandafter\biggr\PartialClose}%
707 \PartialMixSecond%
708 }

```

If the user does not load the `amsmath` package, we will not have access to `\Biggl`, `\Biggr`, `\biggl`, and `\biggr`, so we revert them back to the ordinary `\left` and `\right` versions and warn the user.

```

709 \AtBeginDocument{%
710 \ifpackageloaded{amsmath}{}{}%
711 \PackageWarningNoLine{thermodynamics}
712 {Package amsmath not loaded; load to make PartialBigg and friends
713 work correctly}%
714 \let\PartialBigg\Partial
715 \let\Partialbigg\Partial
716 \let\PartialSecondBigg\PartialSecond
717 \let\PartialSecondbigg\PartialSecond
718 \let\PartialMixSecondBigg\PartialMixSecond
719 \let\PartialMixSecondbigg\PartialMixSecond
720 }%
721 }

```

4.5 Local Override of Delimiters

`thermoparentheses` We define seven environments that *locally* override the delimiters on partial derivatives generated with `\Partial` and friends and/or the subscript notation for partial derivatives.

```

thermobar 722 \newenvironment*{thermoparentheses}{
thermoplain 723 \renewcommand*{\PartialOpen}{(}%
724 \renewcommand*{\PartialClose}{)}%
725 \renewcommand*{\PartialEmptyClose}{})}%
726 }{}
727 \newenvironment*{thermobrackets}{%
728 \renewcommand*{\PartialOpen}{[}%
729 \renewcommand*{\PartialClose}{]}%

```

```

730 \renewcommand*{\PartialEmptyClose}{}}%
731 }{}
732 \newenvironment*{thermobraces}{%
733 \renewcommand*{\PartialOpen}{\}%
734 \renewcommand*{\PartialClose}{\}%
735 \renewcommand*{\PartialEmptyClose}{\}}%
736 }{}
737 \newenvironment*{thermobar}{%
738 \renewcommand*{\PartialOpen}{.}%
739 \renewcommand*{\PartialClose}{\rvert}%
740 \renewcommand*{\PartialEmptyClose}{.}%
741 }{}
742 \newenvironment*{thermoplain}{%
743 \renewcommand*{\PartialOpen}{.}%
744 \renewcommand*{\PartialClose}{.}%
745 \renewcommand*{\PartialEmptyClose}{.}%
746 \@subscriptedfalse
747 }{}
748 \newenvironment*{thermoN0subscripts}{\@subscriptedfalse}{}
749 \newenvironment*{thermosubscripts}{\@subscriptedtrue}{}

```

4.6 User-Interface Macros to Define Symbols

`\DeclareSubscrSymbol` First, we define a (messy!) command that serves to create “subscripted” symbols; for example, typing `\cP_i` should yield C_{P_i} rather than C_{P_i} , C_{P_i} , or C_{P_i} . Superscripts are also handled properly and can be in either order.

```

750 \newcommand*{\DeclareSubscrSymbol}[3]{%
751 \expandafter\def\csname #1\endcsname{%
752 \ifnextchar^{\csname @#1\endcsname}
753 {\ifnextchar_{\csname @#@1\endcsname}{_{#2}_{#3}}}
754 }
755 \expandafter\def\csname @#1\endcsname ##1##2{%
756 {#2}^##2\ifnextchar_{\csname @#@#1\endcsname}{_{#3}}%
757 }
758 \expandafter\def\csname @#@1\endcsname ##1##2{_{#2}_{#3},##2}
759 \expandafter\def\csname @@@#1\endcsname ##1##2{_{#3},##2}
760 }

```

`\heatcapacitiesymbol` Now we define symbols for the heat capacities, compressibilities, and so forth.

```

\compressibilitysymbol 761 \newcommand*{\heatcapacitiesymbol}{C}
\expansivitysymbol    762 \newcommand*{\compressibilitysymbol}{\kappa}
                       763 \newcommand*{\expansivitysymbol}{\alpha}

```

`\cV` The heat capacities are molar by default; we also declare extensive and specific heat
`\cP` capacities. The heat capacities themselves are defined to be “smart”: `\cV_i` will recognize the subscript appropriately and render C_{V_i} rather than C_{V_i} or some other unintended symbol. Superscripts are also handled appropriately and can be in either order.

```

764 \DeclareSubscrSymbol{cV}{\@intensive\heatcapacitiesymbol}{\volume@symbol}
765 \DeclareSubscrSymbol{cP}{\@intensive\heatcapacitiesymbol}{\pressure@symbol}

```

`\cVt` We also introduce *extensive* (rather than molar) heat capacity macros.

`\cPt`

766 `\DeclareSubscrSymbol{cVt}{\@extensive\heatcapacitysymbol}{\volume@symbol}`
767 `\DeclareSubscrSymbol{cPt}{\@extensive\heatcapacitysymbol}{\pressure@symbol}`

`\cPs` `\cPs` and `\cVs` are the specific heat capacities.

`\cVs` 768 `\DeclareSubscrSymbol{cVs}{\@specific\heatcapacitysymbol}{\volume@symbol}`
769 `\DeclareSubscrSymbol{cPs}{\@specific\heatcapacitysymbol}{\pressure@symbol}`

`\kappaT` The isothermal and adiabatic compressibilities are defined similarly, but those do not
`\kappaS` have extensive versions for obvious reasons.
770 `\DeclareSubscrSymbol{kappaT}{\compressibilitysymbol}{\temperature@symbol}`
771 `\DeclareSubscrSymbol{kappaS}{\compressibilitysymbol}{\entropy@symbol}`

`\alphaP` The macro `\alphaP` is intended to refer to the isobaric volume expansivity, while
`\alphaS` `\alphaS` is the isentropic volume expansivity.
772 `\DeclareSubscrSymbol{alphaP}{\expansivitysymbol}{\pressure@symbol}`
773 `\DeclareSubscrSymbol{alphaS}{\expansivitysymbol}{\entropy@symbol}`

`\Psat` The `\Psat` macro (and its clone, the `\Pvap` macro) should be used for the saturation
`\Pvap` pressure.

`\phisat` Similarly, a `\phisat` macro typesets the fugacity coefficient at saturation. The `\fsat`
`\fsat` macro similarly renders the fugacity at saturation. Other saturation properties should
`\sat` use M^{sat} or similar, preferably by defining another macro such as `\Gmsat`.
774 `\newcommand*\sat{{\text{sat}}}`
775 `\newcommand*\Psat{P^{\sat}}`
776 `\let\Pvap\Psat`
777 `\newcommand*\phisat{\phi^{\sat}}`
778 `\newcommand*\fsat{f^{\sat}}`

`std` The `\std` macro denotes standard properties. `\Pstd` and `\fstd` are defined for con-
`Pstd` venience and for consistency across textbooks.
`fstd` 779 `\let\std\circ`
780 `\newcommand*\Pstd{P^{\std}}`
781 `\newcommand*\fstd{f^{\std}}`

`\Deltamix` Mixing properties, such as ΔG_{mix} , should be accessed using `\Deltamix\Gm` and sim-
`\mixing` ilar constructions—this construct will typeset as $\Delta_{\text{MIX}}G$ using the Thompson package
option, for example, and as $\Delta_{\text{mix}}G$ using the Sandler package option.
782 `\newcommand*\mixing{{\text{mix}}}`
783 `\newcommand*\Deltamix}[1]{\Delta{#1}_{\text{mixing}}}`

`\Deltafus` Similar entities for property changes on fusion, reaction, sublimation, and vaporiza-
`\fusion` tion are defined. The heat of reaction is handled differently if `\cP` is its argument: one
`\Deltavap` must typeset $\Delta C_{P,\text{rxn}}$ and $\Delta C_{P,\text{rxn}}^{\circ}$ rather than $\Delta C_{P,\text{rxn}}$ and $\Delta C_{P,\text{rxn}}^{\circ}$.
`\vaporization` 784 `\newcommand*\fusion{{\text{fus}}}`
`\Deltasub` 785 `\newcommand*\reaction{{\text{rxn}}}`
`\sublimation` 786 `\newcommand*\sublimation{{\text{sub}}}`
`\Deltarxn` 787 `\newcommand*\vaporization{{\text{vap}}}`
`\reaction` 788 `\newcommand*\formation{f}`
789 `\newcommand*\Deltafus}[1]{\Delta{#1}^{\fusion}}`
790 `\newcommand*\Deltasub}[1]{\Delta{#1}^{\sublimation}}`
791 `\newcommand*\Deltavap}[1]{\Delta{#1}^{\vaporization}}`

```

792% TODO It would be nice not to have to handle \cP in a special way...
793 \let\Delta@rxn@sym\relax
794 \DeclareSubscrSymbol{@DeltarxncP}{\Delta@rxn@sym}{\pressure@symbol,\reaction}
795 \DeclareSubscrSymbol{@Deltarxn}{\Delta@rxn@sym}{\reaction}
796 \newcommand*{\Deltarxn}[1]{%
797     \def\tmp@arg{#1}%
798     \def\tmp@@arg{\cP}%
799     \ifx\tmp@arg\tmp@@arg\relax
800         \def\@command{\def\Delta@rxn@sym{\Delta\@intensive\heatcapacitiesymbol}%
801             \@DeltarxncP}
802     \else
803         \def\@command{\def\Delta@rxn@sym{\Delta{#1}}\@Deltarxn}
804     \fi
805     \@command
806 }
807 \let\Delta@f@sym\relax
808 \DeclareSubscrSymbol{@Deltaf}{\Delta@f@sym}{f}
809 \newcommand*{\Deltaf}[1]{\def\Delta@f@sym{\Delta{#1}}\@Deltaf}

```

`\fmix` The `\fmix` command is intended to describe fugacities in mixtures. It renders as \hat{f} by default, and would be used as `\fmix_i` or the like, producing \hat{f}_i ; some authors like to use \bar{f}_i or just f_i , and this command creates a consistent way to change between such options.

```
810 \newcommand*{\fmix}{\hat{f}}
```

`\phimix` A similar command, `\phimix`, renders $\hat{\phi}$ by default to represent the fugacity coefficient in the mixture.

```
811 \newcommand*{\phimix}{\hat{\phi}}
```

`\fpure` The `\fpure` command is intended to describe fugacities in pure substances. It renders as f by default, and would be used as `\fpure` or `\fpure_i` or the like, producing f_i ; some authors like to use $f_{\text{pure},i}$, and others like to use f_i^\bullet ; this command creates a consistent way to change between the these options. A similar command for ϕ is given, `\phipure`, for fugacity coefficients.

```
812 \newcommand*{\fpure}{f}
```

```
813 \let\phipure\phi
```

4.7 Partial Molar Quantities

`\partialmolar` Partial molar quantities appear as \bar{G}_i or, for residual properties, \bar{G}_i^R , rather than something like \bar{G}_i or \bar{G}_i^R , which looks better but is harder to implement for obvious reasons. Their definitions allow them to be used as symbols, something like `\Gpm_i`, `\Gpm^{\IGM}_i`, `\Gpm^{\IGM}_i`, and even `\Gpm_i^{\IGM}`; they can also be treated as commands: `\Gpm{i}` is equivalent to `\Gpm_i` and `\Gpm[\IGM]{i}` is equivalent to `\Gpm_i^{\IGM}`. The macro `\partialmolar` can be used to create an arbitrary partial molar symbol.

```
814 \let\pm@symbol\relax
```

```
815 \newcommand*{\partialmolar}[1]{\gdef\pm@symbol{#1}\generic@pm}
```

```
816% if nextchar is _
```

```
817 \def\generic@pm{\@ifnextchar_\@generic@pm_\@generic@pm}
```

```

818% then store the argument
819 \def\@generic@pm#1#2{%
820   \gdef\pm@arg{#2}%
821   % if nextchar after argument is ^
822   \@ifnextchar^\@@@generic@pm\@@@generic@pm
823 }
824% then print it this way
825 \def\@@@generic@pm#1#2{\thermo@overline{\pm@symbol^{#2}_{\pm@arg}}}
826% else do it this way
827 \def\@@@generic@pm{\thermo@overline{\pm@symbol_{\pm@arg}}}
828% else, check whether there is also a ^ coming, otherwise assume
829% it's [] or {}
830 \def\@@@generic@pm{\@ifnextchar^\@@@@generic@pm\pm@bracket@check}
831 \def\@@@@generic@pm#1#2#3#4{\thermo@overline{\pm@symbol^{#2}_{#4}}}
832 \def\pm@bracket@check{\@ifnextchar[\pm@bracket@yes\pm@bracket@no]}
833 \def\pm@bracket@yes[#1]#2{\thermo@overline{\pm@symbol^{#1}_{#2}}}
834 \def\pm@bracket@no#1{\thermo@overline{\pm@symbol_{#1}}}

```

4.8 Symbol Definitions

These macros define the user interface to the symbols for energy, volume, and so forth. We define the “extra” symbols `\Bt` and `\Mt` to represent arbitrary properties.

```

\Nt First, the extensive properties.
\Et 835 \newcommand*{\Nt}{\mole@symbol}
\Ut 836 \newcommand*{\Et}{\@extensive\totalenergy@symbol}
\Ft 837 \newcommand*{\Ut}{\@extensive\internalenergy@symbol}
\Gt 838 \newcommand*{\Ft}{\@extensive\Helmholtz@symbol}
\Ht 839 \newcommand*{\Gt}{\@extensive\Gibbs@symbol}
840 \newcommand*{\Ht}{\@extensive\enthalpy@symbol}
\Lt 841 \newcommand*{\Lt}{\@extensive\Landau@symbol}
\At 842 \newcommand*{\At}{\@extensive\area@symbol}
\St 843 \newcommand*{\St}{\@extensive\entropy@symbol}
\Vt 844 \newcommand*{\Vt}{\@extensive\volume@symbol}
\Qt 845 \newcommand*{\Qt}{\@extensive\heat@symbol}
\Wt 846 \newcommand*{\Wt}{\@extensive\work@symbol}
\Mt 847 \newcommand*{\Mt}{\@extensive M}
\Bt 848 \newcommand*{\Bt}{\@extensive B}

\Em Then the molar properties.
\Um 849 \newcommand*{\Em}{\@intensive\totalenergy@symbol}
\Fm 850 \newcommand*{\Um}{\@intensive\internalenergy@symbol}
\Gm 851 \newcommand*{\Fm}{\@intensive\Helmholtz@symbol}
\Hm 852 \newcommand*{\Gm}{\@intensive\Gibbs@symbol}
853 \newcommand*{\Hm}{\@intensive\enthalpy@symbol}
\Lm 854 \newcommand*{\Lm}{\@intensive\Landau@symbol}
\Am 855 \newcommand*{\Am}{\@intensive\area@symbol}
\Sm 856 \newcommand*{\Sm}{\@intensive\entropy@symbol}
\Vm 857 \newcommand*{\Vm}{\@intensive\volume@symbol}
\Qm 858 \newcommand*{\Qm}{\@intensive\heat@symbol}

\Wm More molar properties.
\Mm 859 \newcommand*{\Wm}{\@intensive\work@symbol}
\Bm

```

```
860 \newcommand*{\Mm}{\@intensive M}
861 \newcommand*{\Bm}{\@intensive B}
```

\Es Now the specific (per unit mass) properties.

```
\Us 862 \newcommand*{\Us}{\@specific\internalenergy@symbol}
\Fs 863 \newcommand*{\Fs}{\@specific\totalenergy@symbol}
\Gs 864 \newcommand*{\Gs}{\@specific\Helmholtz@symbol}
\Hs 865 \newcommand*{\Hs}{\@specific\Gibbs@symbol}
\Ls 866 \newcommand*{\Ls}{\@specific\enthalpy@symbol}
\As 867 \newcommand*{\As}{\@specific\Landau@symbol}
\Ss 868 \newcommand*{\As}{\@specific\area@symbol}
\Ss 869 \newcommand*{\Ss}{\@specific\entropy@symbol}
\Vs 870 \newcommand*{\Vs}{\@specific\volume@symbol}
\Qs 871 \newcommand*{\Qs}{\@specific\heat@symbol}
\Ws 872 \newcommand*{\Ws}{\@specific\work@symbol}
\Ms 873 \newcommand*{\Ms}{\@specific M}
\Bs 874 \newcommand*{\Bs}{\@specific B}
```

\Epm Now the partial molar properties.

```
\Upm 875 \newcommand*{\Epm}{\partialmolar{\totalenergy@symbol}}
\Hpm 876 \newcommand*{\Upm}{\partialmolar{\internalenergy@symbol}}
\Fpm 877 \newcommand*{\Hpm}{\partialmolar{\enthalpy@symbol}}
\Gpm 878 \newcommand*{\Fpm}{\partialmolar{\Helmholtz@symbol}}
\Apm 879 \newcommand*{\Gpm}{\partialmolar{\Gibbs@symbol}}
\Spm 880 \newcommand*{\Apm}{\partialmolar{\area@symbol}}
\Vpm 881 \newcommand*{\Spm}{\partialmolar{\entropy@symbol}}
\Lpm 882 \newcommand*{\Vpm}{\partialmolar{\volume@symbol}}
\Mpm 883 \newcommand*{\Lpm}{\partialmolar{\Omega}}
\Bpm 884 \newcommand*{\Mpm}{\partialmolar{M}}
\Bpm 885 \newcommand*{\Bpm}{\partialmolar{B}}
```

\cPpm Partial molar heat capacities are *hard*, but the following implementation seems to work flawlessly... so far.

```
\cVpm 886 \newcommand*{\cPpm}{%
887 \def\@@@generic@pm##1##2{%
888 \thermo@overline{\pm@symbol^{\##2}_{\pressure@symbol,\pm@arg}}}%
889 \def\@@@generic@pm{\thermo@overline{\pm@symbol_{\pressure@symbol,\pm@arg}}}%
890 \def\@@@generic@pm##1##2##3##4{%
891 \thermo@overline{\pm@symbol^{\##2}_{\pressure@symbol,##4}}}%
892 \def\@pm@bracket@yes[##1]##2{%
893 \thermo@overline{\pm@symbol^{\##1}_{\pressure@symbol,##2}}}%
894 \def\@pm@bracket@no##1{\thermo@overline{\pm@symbol_{\pressure@symbol,##1}}}%
895 \partialmolar{\heatcapacitysymbol}%
896 }
897 \newcommand*{\cVpm}{%
898 \def\@@@generic@pm##1##2{%
899 \thermo@overline{\pm@symbol^{\##2}_{\volume@symbol,\pm@arg}}}%
900 \def\@@@generic@pm{\thermo@overline{\pm@symbol_{\volume@symbol,\pm@arg}}}%
901 \def\@@@generic@pm##1##2##3##4{%
902 \thermo@overline{\pm@symbol^{\##2}_{\volume@symbol,##4}}}%
903 \def\@pm@bracket@yes[##1]##2{%
904 \thermo@overline{\pm@symbol^{\##1}_{\volume@symbol,##2}}}%
905 \def\@pm@bracket@no##1{\thermo@overline{\pm@symbol_{\volume@symbol,##1}}}%
}
```

906 \partialmolar{\heatcapacitysymbol}%
 907 }

4.9 Residual and Excess Properties

\residual Macros are defined for residual properties (departure from non-ideal gases) and
 \excess excess properties (departure from ideal solutions). We begin with two macros to use
 for defining generic residual and excess properties that are not already defined.

908 \newcommand*{\residual}{R}
 909 \newcommand*{\excess}{E}

\UR The ordinary residual properties are molar.

\HR 910 \newcommand*{\UR}{\Um^\residual}
 \FR 911 \newcommand*{\HR}{\Hm^\residual}
 \GR 912 \newcommand*{\FR}{\Fm^\residual}
 \VR 913 \newcommand*{\GR}{\Gm^\residual}
 \SR 914 \newcommand*{\VR}{\Vm^\residual}
 915 \newcommand*{\SR}{\Sm^\residual}

\URt We define similar macros for the extensive residual properties. These are appended
 \HRt by a t (for “total”).

\FRt 916 \newcommand*{\URt}{\Ut^\residual}
 \GRt 917 \newcommand*{\HRt}{\Ht^\residual}
 \VRt 918 \newcommand*{\FRt}{\Ft^\residual}
 \SRt 919 \newcommand*{\GRt}{\Gt^\residual}
 920 \newcommand*{\VRt}{\Vt^\residual}
 921 \newcommand*{\SRt}{\St^\residual}

\URs Similar macros are declared for the specific residual properties. These are appended
 \HRs by an s.

\FRs 922 \newcommand*{\URs}{\Us^\residual}
 \GRs 923 \newcommand*{\HRs}{\Hs^\residual}
 \VRs 924 \newcommand*{\FRs}{\Fs^\residual}
 \SRs 925 \newcommand*{\GRs}{\Gs^\residual}
 926 \newcommand*{\VRs}{\Vs^\residual}
 927 \newcommand*{\SRs}{\Ss^\residual}

\URpm Now for the partial molar residual properties. These are appended by pm.

\HRpm 928 \newcommand*{\URpm}{\partialmolar{\internalenergy@symbol^\residual}}
 \FRpm 929 \newcommand*{\HRpm}{\partialmolar{\enthalpy@symbol^\residual}}
 \GRpm 930 \newcommand*{\FRpm}{\partialmolar{\Helmholtz@symbol^\residual}}
 \VRpm 931 \newcommand*{\GRpm}{\partialmolar{\Gibbs@symbol^\residual}}
 \SRpm 932 \newcommand*{\VRpm}{\partialmolar{\volume@symbol^\residual}}
 933 \newcommand*{\SRpm}{\partialmolar{\entropy@symbol^\residual}}

\UE Now for the excess molar properties.

\HE 934 \newcommand*{\UE}{\Um^\excess}
 \FE 935 \newcommand*{\FE}{\Fm^\excess}
 \GE 936 \newcommand*{\HE}{\Hm^\excess}
 \VE 937 \newcommand*{\GE}{\Gm^\excess}
 \SE 938 \newcommand*{\SE}{\Sm^\excess}
 939 \newcommand*{\VE}{\Vm^\excess}

`\Uet` Similarly, the extensive excess properties.

```

\HET 940 \newcommand*{\Uet}{\Ut^\excess}
\FEt 941 \newcommand*{\FEt}{\Ft^\excess}
\GET 942 \newcommand*{\HEt}{\Ht^\excess}
\VET 943 \newcommand*{\GET}{\Gt^\excess}
\SET 944 \newcommand*{\SEt}{\St^\excess}
      945 \newcommand*{\VET}{\Vt^\excess}

```

`\UES` Now for the specific excess properties.

```

\HES 946 \newcommand*{\UES}{\Us^\excess}
\FES 947 \newcommand*{\FES}{\Fs^\excess}
\GES 948 \newcommand*{\HES}{\Hs^\excess}
\VES 949 \newcommand*{\GES}{\Gs^\excess}
\SES 950 \newcommand*{\SES}{\Ss^\excess}
      951 \newcommand*{\VES}{\Vs^\excess}

```

`\Epm` Finally, the excess partial molar quantities.

```

\HEpm 952 \newcommand*{\UEpm}{\partialmolar{internalenergy@symbol^\excess}}
\FEpm 953 \newcommand*{\HEpm}{\partialmolar{enthalpy@symbol^\excess}}
\GEpm 954 \newcommand*{\FEpm}{\partialmolar{Helmholtz@symbol^\excess}}
\VEpm 955 \newcommand*{\GEpm}{\partialmolar{Gibbs@symbol^\excess}}
\SEpm 956 \newcommand*{\VEpm}{\partialmolar{volume@symbol^\excess}}
      957 \newcommand*{\SEpm}{\partialmolar{entropy@symbol^\excess}}

```

`\prodall` The `\sumall` command and its cousin, `\sumallbutlast`, simplify the typesetting of commonly-used sums; the command `\prodall` does the same thing for products, viz.,

```

\sumall
\sumallbutlast
    \[ \sumall_i x_i = 1 \quad \sumallbutlast_i x_i = 1 - x_{\ncomponents}
      \quad K = \exp\left(\frac{-\Delta G_{rxn}^\circ}{RT}\right)
      \quad = \prodall_i a_i^{\nu_i} \quad \]

```

gives

$$\sum_{i=1}^C x_i = 1 \quad \sum_{i=1}^{C-1} x_i = 1 - x_C \quad K = \exp\left(\frac{-\Delta G_{rxn}^\circ}{RT}\right) = \prod_{i=1}^C a_i^{\nu_i}$$

```

958 \newcommand*{\sumall}{\@ifnextchar_\@sumall\@@sumall}
959 \def\@sumall#1#2{\sum_{#2=1}^{\ncomponents}}
960 \def\@@sumall#1{\sum_{#1=1}^{\ncomponents}}
961 \newcommand*{\sumallbutlast}{\@ifnextchar_\@sumallbutlast\@@sumallbutlast}
962 \def\@sumallbutlast#1#2{\sum_{#2=1}^{\ncomponents-1}}
963 \def\@@sumallbutlast#1{\sum_{#1=1}^{\ncomponents-1}}
964 \newcommand*{\prodall}{\@ifnextchar_\@prodall\@@prodall}
965 \def\@prodall#1#2{\prod_{#2=1}^{\ncomponents}}
966 \def\@@prodall#1{\prod_{#1=1}^{\ncomponents}}

```

`\IG` The `\IG`, `\IGM`, and `\IS` macros (meaning “ideal gas,” “ideal gas mixture,” and “ideal solution,” respectively) should be used to make clean transitions between textbooks—some use “IM” rather than “IS” for example.

```

967 \newcommand*{\IG}{\text{IG}}
968 \newcommand*{\IGM}{\text{IGM}}
969 \newcommand*{\IS}{\text{IS}}

```

`\Henryrat` The Henry's Law constants for the rational basis ($y_i^P = x_i h_i$) and the molal basis
`\Henrymol` ($y_i^P = C_i \mathcal{H}_i$) are given by the macros `\Henryrat` and `\Henrymol`, respectively. Using
 them this way consistently allows for easy switching back and forth.

```
970 \newcommand*{\Henryrat}{h}
971 \newcommand*{\Henrymol}{\mathcal{H}}
```

`\gammarat` The ordinary activity coefficient is universally denoted γ , so I have not defined a
`\gammamol` special macro for that. However, the Henry's Law activity coefficients are far from
 universal, so I have defined macros to make their use consistent. The defaults render
`\gammarat` as γ^* and `\gammamol` as γ^\square . `\gammamol` will use `\square` from packages
 if it is defined; if not, it "fakes it" with the definition below (based on the `amsthm`
 package).

```
972 \AtBeginDocument{%
973   \providecommand*\square{%
974     \text{\leavevmode
975       \hbox to.77778em{%
976         \hfil\vrule
977         \vbox to.6em{\hrule width.6em\vfil\hrule}%
978         \vrule\hfil}%
979     }%
980   }%
981 }
982 \newcommand*\gammarat{\gamma^*}
983 \newcommand*\gammamol{\gamma^\square}
```

`Jacobian` The `\Jacobian` command typesets the Jacobian, viz.,

```
\[ \Jacobian{K}{L}{X}{Y} =
  \begin{vmatrix}
    \displaystyle \Partial{K}{X}{Y} & \displaystyle \Partial{K}{Y}{X} \\
    \displaystyle \Partial{L}{X}{Y} & \displaystyle \Partial{L}{Y}{X}
  \end{vmatrix}
\]
```

produces

$$\begin{pmatrix} \partial(K,L) \\ \partial(X,Y) \end{pmatrix} = \begin{vmatrix} \left(\frac{\partial K}{\partial X}\right)_Y & \left(\frac{\partial K}{\partial Y}\right)_X \\ \left(\frac{\partial L}{\partial X}\right)_Y & \left(\frac{\partial L}{\partial Y}\right)_X \end{vmatrix}$$

This macro is defined purely for convenience.

```
984 \newcommand*\Jacobian[4]{\frac{\partial\{#1,#2\}}{\partial\{#3,#4\}}}
```

Change History

v1.00

General: Initial public release 1

Index

Numbers written in *italics* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

Symbols	
<code>\@extensive</code>	<u>21</u>
<code>\@intensive</code>	<u>21</u>
<code>\@specific</code>	<u>21</u>
A	
<code>\allbut</code>	<i>11</i> , <u>23</u>
<code>\allbutlastand</code> ...	<i>11</i> , <u>23</u>
<code>\allcomponents</code>	<u>23</u>
<code>\allMs</code>	<i>10</i> , <u>23</u>
<code>\allMsbut</code>	<i>10</i> , <u>23</u>
<code>\allmus</code>	<i>10</i> , <u>23</u>
<code>\allmusbut</code>	<i>10</i> , <u>23</u>
<code>\allNs</code>	<i>10</i> , <i>15</i> , <i>16</i> , <u>23</u>
<code>\allNsbut</code> ..	<i>10</i> , <i>15</i> , <i>16</i> , <u>23</u>
<code>\allWs</code>	<i>10</i> , <u>23</u>
<code>\allWsbu</code>	<i>10</i> , <u>23</u>
<code>\allXs</code>	<i>10</i> , <i>16</i> , <u>23</u>
<code>\allXsbu</code>	<i>10</i> , <i>16</i> , <u>23</u>
<code>\allYs</code>	<i>10</i> , <u>23</u>
<code>\allYsbu</code>	<i>10</i> , <u>23</u>
<code>\alphaP</code>	<i>4</i> , <i>5</i> , <u>36</u>
<code>\alphaS</code>	<i>4</i> , <i>5</i> , <u>36</u>
<code>\Am</code>	<i>4</i> , <u>38</u>
<code>\Apm</code>	<i>4</i> , <u>39</u>
<code>\area@symbol</code>	<u>19</u>
<code>\As</code>	<i>4</i> , <u>39</u>
<code>\At</code>	<i>4</i> , <u>38</u>
B	
<code>\Bm</code>	<i>4</i> , <u>38</u>
<code>\Bpm</code>	<i>4</i> , <u>39</u>
<code>\Bs</code>	<i>4</i> , <u>39</u>
<code>\Bt</code>	<i>4</i> , <u>38</u>
C	
<code>\compressibilitysymbol</code>	<i>5</i> , <u>35</u>
<code>\cP</code>	<i>4</i> , <i>5</i> , <u>35</u>
<code>\cPpm</code>	<i>3</i> , <u>39</u>
<code>\cPs</code>	<i>3</i> , <u>36</u>
<code>\cPt</code>	<i>3</i> , <u>35</u>
<code>\cV</code>	<i>4</i> , <i>5</i> , <u>35</u>
<code>\cVpm</code>	<i>3</i> , <u>39</u>
<code>\cVs</code>	<i>3</i> , <u>36</u>
<code>\cVt</code>	<i>3</i> , <u>35</u>
D	
<code>\dbar</code>	<i>15</i> , <u>18</u>
<code>\DeclareSubscrSymbol</code> .	<u>35</u>
<code>\Deltaf</code>	<u>7</u>
<code>\Deltafus</code>	<i>5</i> , <i>7</i> , <u>36</u>
<code>\Deltamix</code>	<i>5</i> , <i>7</i> , <u>36</u>
<code>\Deltarxn</code>	<i>5</i> , <i>7</i> , <u>36</u>
<code>\Deltasub</code>	<i>5</i> , <i>7</i> , <u>36</u>
<code>\Deltavap</code>	<i>5</i> , <i>7</i> , <u>36</u>
E	
<code>\Em</code>	<i>4</i> , <u>38</u>
<code>\enthalpy@symbol</code>	<u>19</u>
<code>\entropy@symbol</code>	<u>19</u>
environments:	
thermobar	<i>15</i> , <u>34</u>
thermobraces	<u>34</u>
thermobrackets	<i>15</i> , <u>34</u>
thermoNOsubscripts	<u>15</u>
thermoparentheses	<i>15</i> , <u>34</u>
thermoplain ...	<i>15</i> , <u>34</u>
thermosubscripts ..	<u>15</u>
<code>\Epm</code>	<i>4</i> , <u>39</u>
<code>\Es</code>	<i>4</i> , <u>39</u>
<code>\Et</code>	<i>4</i> , <u>38</u>
<code>\excess</code>	<i>8</i> , <u>40</u>
<code>\expansivitysymbol</code> ..	<i>5</i> , <u>35</u>
F	
<code>\FE</code>	<u>40</u>
<code>\FEpm</code>	<u>41</u>
<code>\FEs</code>	<u>41</u>
<code>\FEt</code>	<u>41</u>
<code>\Fm</code>	<i>4</i> , <u>38</u>
<code>\fmix</code>	<i>5</i> , <i>6</i> , <u>37</u>
<code>\Fpm</code>	<i>4</i> , <u>39</u>
<code>\fpure</code>	<i>5</i> , <i>6</i> , <u>37</u>
<code>\FR</code>	<u>40</u>
<code>\FRpm</code>	<u>40</u>
<code>\FRs</code>	<u>40</u>
<code>\FRt</code>	<u>40</u>
<code>\Fs</code>	<i>4</i> , <u>39</u>
<code>\fsat</code>	<i>5</i> , <i>6</i> , <u>36</u>
<code>\fstd</code>	<i>7</i> , <u>36</u>
<code>\Ft</code>	<i>4</i> , <u>38</u>
<code>\fusion</code>	<i>5</i> , <u>36</u>
<code>\gammamat</code>	<i>5</i> , <i>6</i> , <u>42</u>
<code>\GE</code>	<u>40</u>
<code>\GEpm</code>	<u>41</u>
<code>\GEs</code>	<u>41</u>
<code>\GEt</code>	<u>41</u>
<code>\Gibbs@symbol</code>	<u>19</u>
<code>\Gm</code>	<i>4</i> , <u>38</u>
<code>\Gpm</code>	<i>4</i> , <u>39</u>
<code>\GR</code>	<u>40</u>
<code>\GRpm</code>	<u>40</u>
<code>\GRs</code>	<u>40</u>
<code>\GRt</code>	<u>40</u>
<code>\Gs</code>	<i>4</i> , <u>39</u>
<code>\Gt</code>	<i>4</i> , <u>38</u>
H	
<code>\HE</code>	<u>40</u>
<code>\heat@symbol</code>	<u>19</u>
<code>\heatcapacitysymbol</code> ..	<i>5</i> , <u>35</u>
<code>\Helmholtz@symbol</code> ...	<u>19</u>
<code>\Henrymol</code>	<i>5</i> , <i>6</i> , <u>42</u>
<code>\Henryrat</code>	<i>5</i> , <i>6</i> , <u>42</u>
<code>\HEpm</code>	<u>41</u>
<code>\HEs</code>	<u>41</u>
<code>\HEt</code>	<u>41</u>
<code>\Hm</code>	<i>3</i> , <i>4</i> , <u>38</u>
<code>\Hpm</code>	<i>3</i> , <i>4</i> , <u>39</u>
<code>\HR</code>	<u>40</u>
<code>\HRpm</code>	<u>40</u>
<code>\HRs</code>	<u>40</u>
<code>\HRT</code>	<u>40</u>
<code>\Hs</code>	<i>3</i> , <i>4</i> , <u>39</u>
<code>\Ht</code>	<i>3</i> , <i>4</i> , <u>38</u>
I	
<code>\IG</code>	<u>41</u>
<code>\IGM</code>	<u>41</u>
<code>\internalenergy@symbol</code>	<u>19</u>
<code>\IS</code>	<u>41</u>
J	
<code>\Jacobian</code>	<u>42</u>
K	
<code>\kappaS</code>	<i>4</i> , <i>5</i> , <u>36</u>
<code>\kappaT</code>	<i>4</i> , <i>5</i> , <u>36</u>
<code>\gammamol</code>	<i>5</i> , <i>6</i> , <u>42</u>

L		<code>\phimix</code> 5, 6, <u>37</u>	<code>thermoN0subscripts</code> (environment) 15
<code>\Landau@symbol</code> 19		<code>\phipure</code> 5, <u>6</u>	<code>thermoparentheses</code> (environment) . . 15, <u>34</u>
<code>\Lm</code> 4, <u>38</u>		<code>\phisat</code> 5, 6, <u>36</u>	<code>thermoplain</code> (environment) 15, <u>34</u>
<code>\Lpm</code> 4, <u>39</u>		<code>\pressure@symbol</code> 19	<code>thermosubscripts</code> (environment) 15
<code>\Ls</code> 4, <u>39</u>		<code>\prodall</code> 11, <u>41</u>	<code>\totalenergy@symbol</code> . 19
<code>\Lt</code> 4, <u>38</u>		<code>\Psat</code> 5, 6, <u>36</u>	
M		<code>\Pstd</code> 7, <u>36</u>	
<code>\mixing</code> 5, 7, <u>36</u>		<code>\Pvap</code> 5, 6, <u>36</u>	
<code>\Mm</code> 4, <u>38</u>		Q	
<code>\mole@symbol</code> 19		<code>\Qm</code> 4, <u>38</u>	U
<code>\Mpm</code> 4, <u>39</u>		<code>\Qs</code> 4, <u>39</u>	<code>\UE</code> 8, <u>40</u>
<code>\Ms</code> 4, <u>39</u>		<code>\Qt</code> 4, <u>38</u>	<code>\UEpm</code> 8, <u>41</u>
<code>\Mt</code> 4, <u>38</u>		R	
N		<code>\reaction</code> 5, <u>36</u>	<code>\UES</code> 8, <u>41</u>
<code>\ncomponents</code> 15, <u>23</u>		<code>\residual</code> 8, <u>40</u>	<code>\UET</code> 8, <u>41</u>
<code>\Nt</code> 4, <u>38</u>		S	
P		<code>\sat</code> 5, 6, <u>36</u>	<code>\Um</code> 3, 4, <u>38</u>
<code>\Partial</code> 9, <u>29</u>		<code>\SE</code> 40	<code>\Upm</code> 3, 4, <u>39</u>
<code>\Partial*</code> 9, <u>29</u>		<code>\SEpm</code> 41	<code>\UR</code> 8, <u>40</u>
<code>\PartialBigg</code> 9, <u>31</u>		<code>\SEs</code> 41	<code>\URpm</code> 8, <u>40</u>
<code>\Partialbigg</code> 9, <u>31</u>		<code>\SEt</code> 41	<code>\URs</code> 8, <u>40</u>
<code>\PartialBigg*</code> 9, <u>31</u>		<code>\Sm</code> 4, <u>38</u>	<code>\URt</code> 8, <u>40</u>
<code>\Partialbigg*</code> 9, <u>31</u>		<code>\Spm</code> 4, <u>39</u>	<code>\Us</code> 4, <u>39</u>
<code>\PartialClose</code> 19		<code>\SR</code> 40	<code>\Ut</code> 3, 4, <u>38</u>
<code>\PartialEmptyClose</code> . . 19		<code>\SRpm</code> 40	
<code>\PartialMixSecond</code> . 9, <u>32</u>		<code>\SRs</code> 40	V
<code>\PartialMixSecond*</code> 9, <u>32</u>		<code>\SRt</code> 40	<code>\vaporization</code> 5, <u>36</u>
<code>\PartialMixSecondBigg</code>		<code>\Ss</code> 4, <u>39</u>	<code>\VE</code> 40
. 9, <u>34</u>		<code>\St</code> 4, <u>38</u>	<code>\VEpm</code> 41
<code>\PartialMixSecondbigg</code>		<code>\std</code> 7, <u>36</u>	<code>\VEs</code> 41
. 9, <u>34</u>		<code>\sublimation</code> 5, <u>36</u>	<code>\VET</code> 41
<code>\PartialMixSecondBigg*</code>		<code>\sumall</code> 11, <u>41</u>	<code>\Vm</code> 38
. 9, <u>34</u>		<code>\sumallbutlast</code> . . . 11, <u>41</u>	<code>\volume@symbol</code> 19
<code>\PartialMixSecondbigg*</code>		T	
. 9, <u>34</u>		<code>\temperature@symbol</code> . 19	<code>\Vpm</code> 39
<code>\partialmolar</code> 4, <u>37</u>		<code>\thermo@overline</code> . . . 19	<code>\VR</code> 40
<code>\PartialOpen</code> 19		<code>\thermo@underline</code> . . 19	<code>\VRpm</code> 40
<code>\PartialSecond</code> 9, <u>31</u>		<code>thermobar</code> (environment)	<code>\VRs</code> 40
<code>\PartialSecond*</code> . . . 9, <u>31</u>	 15, <u>34</u>	<code>\VRT</code> 40
<code>\PartialSecondBigg</code> 9, <u>32</u>		<code>thermobraces</code> (environment)	<code>\Vs</code> 39
<code>\PartialSecondbigg</code> 9, <u>32</u>	 34	<code>\Vt</code> 38
<code>\PartialSecondBigg*</code> 9, <u>32</u>		<code>thermobrackets</code> (environment)	
<code>\PartialSecondbigg*</code> 9, <u>32</u>	 15, <u>34</u>	W