

The thermodynamics package*

Karl D. Hammond
hammondkd@missouri.edu

2023/12/05

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, particularly 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 v2.01`, dated 2023/12/05.

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

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

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 change notation temporarily.

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

`\Ht` shown in Table 1. Examples using the enthalpy are

`\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.

`\cP` The heat capacities (see Table 2) are generally assumed to be molar (e.g., `\cP`
`\cV` is interpreted to be the molar heat capacity). To get the specific heat capacities,
`\cPt` the macros `\cPs` and `\cVs` are provided, which by default render as \hat{C}_P and \hat{C}_V ,
`\cVt` respectively. There are also extensive versions, so `\cPt` and `\cVt` will render as \underline{C}_P
`\cPs` and \underline{C}_V , respectively. Note that `\cP` and friends require you to surround `\text` with
`\cVs` brackets. For example, `\cP_i^\text{A}` will not work, nor will `\cP^\text{A}_i`; you
`\cPpm` need to use `\cP_i^\{\text{A}\}` and `\cP^\{\text{A}\}_i`. Using `\cP^\IG_i` will work
`\cVpm` 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:

$$\begin{aligned} \backslash[\cVpm_i &= \text{Partial}^*\{\cVt\}\{\Nt_i\}\{T,P,\text{allNsbut}\{i\}\} \\ &= \frac{\text{partial}\{\text{partial}\Nt_i\}}{\text{left}[-T\text{PartialSecond}\{Ft\}\{T\}\{\Vt,\text{allNs}\} \\ &\quad \text{right}\}_{T,P,\text{allNsbut}\{i\}\}} \backslash] \end{aligned}$$

yields

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

2.2 Partial Molar Properties

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

`\[\Upm{i} \quad \Upm[\IG]{i} \quad \Vpm_i \quad \Vpm^\IG_i \]`

will typeset as

$$\overline{U}_i \quad \overline{U}_i^{\text{IG}} \quad \overline{V}_i \quad \overline{V}_i^{\text{IG}}.$$

There are also partial molar heat capacities available via the macros `\cPpm` and `\cVpm`. **Important:** The `\text` command defined by the `amstext` package is usually robust enough that something like `\Um^\text{L}` will work as expected, without additional

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	Excess				Residual (Departure)				
					T	M	S	PM	T	M	S	PM	
Heat	<code>\Qt</code>	<code>\Qm</code>	<code>\Qs</code>										
Work	<code>\Wt</code>	<code>\Wm</code>	<code>\Ws</code>										
Total energy	<code>\Et</code>	<code>\Em</code>	<code>\Es</code>	<code>\Epm</code>	<code>\EEt</code>	<code>\EE</code>	<code>\EEs</code>	<code>\EEpm</code>	<code>\ERt</code>	<code>\ER</code>	<code>\ERs</code>	<code>\ERpm</code>	
Internal energy	<code>\Ut</code>	<code>\Um</code>	<code>\Us</code>	<code>\Upm</code>	<code>\UEt</code>	<code>\UE</code>	<code>\UES</code>	<code>\UEpm</code>	<code>\URt</code>	<code>\UR</code>	<code>\URs</code>	<code>\URpm</code>	
Enthalpy	<code>\Ht</code>	<code>\Hm</code>	<code>\Hs</code>	<code>\Hpm</code>	<code>\HEt</code>	<code>\HE</code>	<code>\HEs</code>	<code>\HEpm</code>	<code>\HRt</code>	<code>\HR</code>	<code>\HRs</code>	<code>\HRpm</code>	
Entropy	<code>\St</code>	<code>\Sm</code>	<code>\Ss</code>	<code>\Spm</code>	<code>\SEt</code>	<code>\SE</code>	<code>\SEs</code>	<code>\SEpm</code>	<code>\SRt</code>	<code>\SR</code>	<code>\SRs</code>	<code>\SRpm</code>	
Volume	<code>\Vt</code>	<code>\Vm</code>	<code>\Vs</code>	<code>\Vpm</code>	<code>\VET</code>	<code>\VE</code>	<code>\VES</code>	<code>\VEpm</code>	<code>\VRt</code>	<code>\VR</code>	<code>\VRs</code>	<code>\VRpm</code>	
Helmholtz free energy	<code>\Ft</code>	<code>\Fm</code>	<code>\Fs</code>	<code>\Fpm</code>	<code>\FEt</code>	<code>\FE</code>	<code>\FEs</code>	<code>\FEpm</code>	<code>\FRt</code>	<code>\FR</code>	<code>\FRs</code>	<code>\FRpm</code>	
Gibbs free energy	<code>\Gt</code>	<code>\Gm</code>	<code>\Gs</code>	<code>\Gpm</code>	<code>\GET</code>	<code>\GE</code>	<code>\GES</code>	<code>\GEpm</code>	<code>\GRt</code>	<code>\GR</code>	<code>\GRs</code>	<code>\GRpm</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>	<code>\LEt</code>	<code>\LE</code>	<code>\LEs</code>	<code>\LEpm</code>	<code>\LRt</code>	<code>\LR</code>	<code>\LRs</code>	<code>\LRpm</code>	
Moles	<code>\Nt</code>												

^aThe grand potential, $\underline{\Omega}(T, \underline{V}, \mu_1, \dots, \mu_C) = \underline{U} - T\underline{S} - \sum_{i=1}^C \mu_i n_i$, is also called the Landau free energy by some authors.

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 effectively defined via the macro

```
\NewDocumentCommand{\Gpm}{}{\partialmolar{G}}.
```

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 Defining New Properties

`\NewExtensiveProperty` Users can create new properties using a family of commands. Typically, a user would want to define at least the total, molar, and specific properties, which can be accomplished by the `\NewExtensiveProperty` command. For example,

```
\NewExtensiveProperty{J}{K}
```

would define the commands `\Jt`, `\Jm`, and `\Js`, which would produce (using the default package options) the symbols \underline{K} , K , and \hat{K} , respectively.

`\NewPartialMolarProperty` Partial molar properties can be created either with the `\partialmolar` macro directly as described above or with `\NewPartialMolarProperty`, which has the same argument style as `\NewExtensiveProperty`.

`\NewExcessProperty` Similarly, one can define commands for the total, molar, and specific excess properties using `\NewExcessProperty` in a similar manner, and similar commands for the residual properties with `\NewResidualProperty`.

`\NewThermodynamicProperty` It is common that a user wants the total, molar, specific, and partial molar commands for a new symbol, as well as excess and residual (departure) properties for each case. Users can define such properties—common examples are \underline{B} and \underline{M} to represent

Table 2. Convenience macros and their default symbols. These are generally “smart”: for example, `\cPi` renders as $C_{P,i}$, as expected, and `\cPistd` renders as $C_{P,i}^{\circ}$, also as expected. You can also reverse it: `\cPstdi` 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>
Joule–Thomson coeff.	<code>\mu_{JT}</code>	μ_{JT}	$\left(\frac{\partial T}{\partial P} \right)_H$	<code>\JTsymbol</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 coefficient	<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}	$M - \sum_{i=1}^C x_i M_i$	<code>\mixing</code>
Change on reaction	<code>\Deltarxn^d</code>	ΔM_{rxn}	$\sum_{i=1}^C \nu_i M_i$	<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 `\mui = \mustdi + RT \log ai`, yielding $\mu_i = \mu_i^{\circ} + RT \log a_i$.

^dThe usual usage would be something like `\Deltamix \VmIGM = 0$`.

uncommon or arbitrary properties—by using `\NewThermodynamicProperty`, which calls all four of the aforementioned declarations on the same command/symbol combinations. For example,

```
\NewThermodynamicProperty{B}{B}
```

defines the commands `\Bt`, `\Bm`, `\Bs`, and `\Bpm`, which define the total, molar, specific, and partial molar properties, respectively. It also defines `\BEt`, `\BE`, `\BEs`, and `\BEpm` for the corresponding excess properties, and `\BRt`, `\BR`, `\BRs`, and `\BRpm` for residual (departure) properties. These produce, respectively, \underline{B} , B , \hat{B} , \bar{B}_i , \underline{B}^E , B^E , \hat{B}^E , \bar{B}_i^E , \underline{B}^R , B^R , \hat{B}^R , and \bar{B}_i^R using the defaults.

2.4 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.

2.4.1 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; 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.

2.4.2 Joule–Thomson Coefficients

`\muJT` The Joule–Thomson coefficient is produced with `\muJT`, which by default is rendered μ_{JT} . Some books call this coefficient α_H ; this is handled automatically for books of which the package author is aware.

2.4.3 Fugacities and Fugacity Coefficients

`\fpure` Different textbooks use different variations on 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:

$$\left[\begin{aligned} \text{\fmix}_j &= x_j \text{\phimix}_j P = x_j \gamma_j \text{\fpure}_j \\ &= x_j \gamma_j \text{\hipure}_j P. \end{aligned} \right]$$

With the default package options, this produces

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

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

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

Similarly, the Prausnitz package option causes it to generate

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

and the Sandler option causes it to generate

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

2.4.4 Activity Coefficients and Henry's Constants

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

$$\begin{aligned} \left[\text{\fmix}_i = x_i \gamma_i \text{\fpure}_i = x_i \text{\gammarat}_i \text{\Henryrat}_i \right. \\ \left. = C_i \text{\gammamol}_i \text{\Henrymol}_i \right] \end{aligned}$$

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. These symbols can be customized either directly or by using package options. For example, the `TesterModel1` package option changes the equation above to

$$\hat{f}_i = x_i \gamma_i f_i = x_i \gamma_i^* f_i^{**} = C_i \gamma_i^* f_i^*$$

without any changes in markup.

Note that some books (e.g., `ElliottLira`) prefer the atmospheric chemistry convention for the molal basis, namely

$$\hat{f}_i = x_i \gamma_i f_i = x_i \gamma_i^* h_i = C_i \gamma_i^\square / K_{H,i}$$

(that is, using $1/K_{H,i}$ in place of \mathcal{H}_i). Unfortunately, this makes it impossible to change symbols without any edits to markup, so this convention is *not* supported by this package.

2.4.5 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 `\phisat` 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 (using the defaults). If you wanted to redefine them to be P^{vap} , f^{vap} , and ϕ^{vap} , you could simply redefine `\sat` with `\RenewDocumentCommand{\sat}{}{\{\text{vap}\}}`. This is done automatically using package option `Sandler`.

2.4.6 Standard States

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 `\std` macro. The macro `\Pstd` is defined as P° 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:

$$\begin{aligned} \Delta G_{\text{rxn}} = \sum_i \nu_i \mu_i &= \sum_i \left[\nu_i \mu_i^{\text{std}} + \nu_i RT \log \left(\frac{f_{\text{mix}_i}}{f_{\text{std}_i}} \right) \right] \\ &= \Delta G_{\text{rxn}}^{\text{std}} + RT \log \left[\prod_i a_i^{\nu_i} \right], \end{aligned}$$

which produces

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

with the default settings.

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

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

$$\Delta G_{\text{mix}} = G - \sum_i x_i G_i = \Delta H_{\text{mix}} - T \Delta S_{\text{mix}}$$

which yields, using the default options,

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

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}`.

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

Their use is straightforward, viz.,

$$\begin{aligned} \Delta H_{\text{sub}} &= H^{\text{V}} - H^{\text{S}} = H^{\text{V}} - H^{\text{L}} + (H^{\text{L}} - H^{\text{S}}) \\ &= \Delta H_{\text{fus}} + \Delta H_{\text{vap}} \end{aligned}$$

yielding

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

with the default options. Note that some textbooks (e.g., Sandler) typeset these quantities quite differently; this is handled automatically. Other textbooks (e.g., Koretsky) typeset them as subscripts; this is also handled automatically.

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

$$\Delta G_{\text{rxn}}^{\text{std}} = \sum_i \nu_i \Delta f_{\text{H}_i}^{\text{std}}$$

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.

2.5 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

```
\RenewDocumentCommand{\excess}{}{\{EX\}},
```

or possibly

```
\RenewDocumentCommand{\excess}{}{\{\mathrm{EX}\}}
```

or even

```
\RenewDocumentCommand{\excess}{}{\text{EX}},
```

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

It is generally possible to use superscripts with the excess or residual properties; in the event this fails, the `\excess` and `\residual` macros can be used directly, viz.,

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

which yields

$$H^{E,\circ} = H^{E,\circ} = H^E(T, P^\circ) = H^R(T, P^\circ) - H^{R,IS}(T, P^\circ) = H^{R,\circ} - H^{R,IS,\circ}$$

$$H^{EX,\circ} = H^{EX,\circ} = H^{EX}(T, P^\circ) = H^R(T, P^\circ) - H^{R,IS}(T, P^\circ)$$

using the default options.

2.6 Partial Derivatives

`\Partial` Partial derivatives are easily rendered using the `\Partial` command. There is a `\Partial*` 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:

```
\[ \Partial{\Hm}{T}{P} = \cP \quad \Partial*{\Hm}{T}{P} = \cP \],
```

which yields

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

`\Partialinline` Inline first derivatives² can be entered the same way; compare:

```
\[ \Partial*{\Hm}{T}{P} = \Partialinline{\Hm}{T}{P}
= T\Partial{\Sm}{T}{P} \],
```

which results in

$$\left(\frac{\partial H}{\partial T}\right)_P = (\partial H/\partial T)_P = T(\partial S/\partial T)_P.$$

There is no need for an inline starred form, as the subscripts do not extend far enough below the baseline.

2.6.1 Second-Order Partial Derivatives

`\PartialSecond` Second partial derivatives and mixed-second partial derivatives are typeset with the `\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

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

`\PartialSecondinline` Inline versions² of second-order derivatives are handled with `\PartialSecondinline` and `\PartialMixSecondinline`, viz.,

```
\begin{equation}
\ \Vpm_i = \Partialinline{\Vt}{\Nt_i}{T,P,\allNsbut{i}}
= \PartialMixSecondinline{\Gt}{\Nt_i}{P}{T,\allNsbut{i}}
= \PartialMixSecondinline{\Gt}{P}{\Nt_i}{T,\allNsbut{i}}
= \Partialinline{\Gpm_i}{P}{T,\allNs} \ \
\end{equation}
```

¹You knew that joke was coming.

²The “inline” versions of partial derivatives are “sticky”: if you issue `\Partialinline` or its second-order equivalents anywhere in a line, all subsequent `\Partial` and `\Partial[Mix]Second` macros on the same line (technically, anywhere in the same \TeX “group”) will expand inline as well. To prevent this, enclose your `\Partialinline` and associated arguments in its own group (i.e., `\Partialinline{...}`).

looks like

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

2.6.2 Delimiter Sizing

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 delimiters 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

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

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

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

Similarly, there are times when `\Partialinline` causes parentheses that are too big for inline text, and they do not need to be—particularly for symbols with overlines, underlines, or other decorations. In this case, the macros `\Partialinlinetext`, `\PartialSecondinlinetext`, and `\PartialMixSecondinlinetext` come in handy; using these macros like so,

```
\[ \Vpm_i = \Partialinlinetext{Vt}{Nt_i}{T,P,\allNsbut{i}}
      = \PartialMixSecondinlinetext{Gt}{Nt_i}{P}{T,\allNsbut{i}}
      = \PartialMixSecondinlinetext{Gt}{P}{Nt_i}{T,\allNsbut{i}}
      = \Partialinlinetext{Gpm_i}{P}{T,\allNs} \]
```

Equation (2) looks like

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

2.6.3 Higher-Order Derivatives

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

$$\begin{aligned} \backslash[\backslash cPpm_i &= T\backslash\text{Partial}^*\{\backslash\text{Spm}_i\}\{T\}\{P, \backslash\text{allXs}\} \\ &= T\backslash\text{PartialMixSecond}^*\{\backslash\text{St}\}\{T\}\{\backslash\text{Nt}_i\}\{P, \backslash\text{allNsbut}\{i\}\} \\ &= -T\backslash\text{Partial}\{\wedge^3\backslash\text{Gt}\}\{T^2\backslash\text{partial}\backslash\text{Nt}_i\}\{P, \backslash\text{allNsbut}\{i\}\} \backslash] \end{aligned}$$

gives

$$\overline{C_{P,i}} = T \left(\frac{\partial \overline{S_{P,i}}}{\partial T} \right)_{P, \vec{x}} = T \left(\frac{\partial^2 \underline{S}}{\partial T \partial n_i} \right)_{P, n_{j \neq i}} = -T \left(\frac{\partial^3 \underline{G}}{\partial T^2 \partial n_i} \right)_{P, n_{j \neq i}},$$

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

2.7 Holding Constant the Number of Moles of Several Species

`\allNs` It is common in thermodynamics to use notation such as
`\allNsbut`
`\allmus`
`\allmusbut`
`\allXs`
`\allXsbut` or perhaps
`\allYs`
`\allYsbut`
`\allMs` or
`\allMsbut`
`\allWs`
`\allWsbut`

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

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

$$\overline{V}_k = \left(\frac{\partial \underline{V}}{\partial n_k} \right)_{T, P, n_{[k]}}$$

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

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

or perhaps

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

There are several macros that standardize such constructs. The `\allNs` macro expands to something meaning the number of moles of all species; by default, this is \vec{n} (package option `moles-index`), but can be changed to n_1, \dots, n_C using the package option `moles-range`. Similarly, the macros `\allmus` and `\allmusbut` do the same but with n replaced by μ , and `\allMsbut` is the same with m instead of n . There are analogous macros for mole fractions, namely `\allXs` and `\allXsbut` for x and `\allYs` and `\allYsbut` for y , as well as `\allWs` and `\allWsbut` for mass fractions—these implicitly assume that all mole or mass fractions *except* the last are used as variables. These macros all take an optional argument; for example,

$$\backslash[\backslash\text{Partial}\{\backslash\text{Ht}\}\{\backslash\text{Nt}_1\}\{T, P, \backslash\text{allNsbut}\{1\}\} = \backslash\text{Partial}\{\backslash\text{Ht}\}\{\backslash\text{Nt}_1\}\{T, P, \backslash\text{allNsbut}[m]\{1\}\} = \backslash\text{Hpm}_1 \backslash]$$

typesets as

$$\left(\frac{\partial H}{\partial n_1}\right)_{T,P,n_{j\neq 1}} = \left(\frac{\partial H}{\partial n_1}\right)_{T,P,n_{m\neq 1}} = \overline{H}_1.$$

Similarly,

$$\begin{aligned} & \left[\frac{\partial H}{\partial x_i} \right]_{T,P,\text{allXsbut}\{i\}} \\ &= \left[\frac{\partial H}{\partial x_i} \right]_{T,P,\text{allXsbut}\{m\}\{i\}} \\ &= H_{pm_i} - H_{pm_ncomponents} \end{aligned}$$

becomes

$$\left(\frac{\partial H}{\partial x_i}\right)_{T,P,x_{j\neq i,C}} = \left(\frac{\partial H}{\partial x_i}\right)_{T,P,x_{m\neq i,C}} = \overline{H}_i - \overline{H}_C.$$

Users must supply their own redefinitions if they wish to hold something other than `\ncomponents` constant in addition to the argument for mole and mass fractions. Using the `moles-range` package option, for which `\allXsbut{k}` expands to $x_1, \dots, [x_k], \dots, x_{C-1}$ rather than $x_{j\neq k,C}$, the optional argument is ignored.

The optional argument to `\allNs` and similar commands is ignored when using the default options; it is relevant for package options that redefine `\allNs` to make N_i , for example; in this case, one can enter `\allNs[j]` to make TeX render N_j instead of N_i . This is useful if you are using i somewhere else in the equation.

`\allbut` Users can define new “all but” macros using the `\allbut` and `\allbutlastand`
`\allbutlastand` commands. For example,

```
\NewDocumentCommand{\allNsbut}{0{j} m}{\allbut[#1]{#2}\Nt}
\NewDocumentCommand{\allXsbut}{0{j} m}{\allbutlastand[#1]{#2}{x}}
```

are the definitions of `\allNsbut` and `\allXsbut`, respectively.

2.8 Jacobians

`\Jacobian` The Jacobian determinant is often denoted with Leibnitz-like notation, viz.,
`\Jacobidetet` $\left[\frac{\partial(K,L)}{\partial(X,Y)} \right] = \text{Jacobidetet}\{K,L\}\{X,Y\}$,

which produces (assuming the `amsmath` package has been loaded)

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

There are two optional arguments to `\Jacobidetet`. The first will be pre-pended before every element of the matrix (typically `\textstyle` or `\displaystyle`); the second is the extra spacing added between rows (default is 1.25 ex for text-style fractions and 2.75 ex for display-style fractions). More than two variables can be specified, viz.,

```
\left[ \frac{\partial(K,L,M)}{\partial(X,Y,Z)} \right]
= \text{Jacobidetet}[\displaystyle][3ex]\{K,L,M\}\{X,Y,Z\}
```

will produce

$$\frac{\partial(K, L, M)}{\partial(X, Y, Z)} = \begin{vmatrix} \left(\frac{\partial K}{\partial X}\right)_{Y,Z} & \left(\frac{\partial K}{\partial Y}\right)_{X,Z} & \left(\frac{\partial K}{\partial Z}\right)_{X,Y} \\ \left(\frac{\partial L}{\partial X}\right)_{Y,Z} & \left(\frac{\partial L}{\partial Y}\right)_{X,Z} & \left(\frac{\partial L}{\partial Z}\right)_{X,Y} \\ \left(\frac{\partial M}{\partial X}\right)_{Y,Z} & \left(\frac{\partial M}{\partial Y}\right)_{X,Z} & \left(\frac{\partial M}{\partial Z}\right)_{X,Y} \end{vmatrix}.$$

The `\Jacobiandet` macro will understand implied multicomponent Jacobians, too, namely,

```
\[ \Jacobian{f_1,\dots,f_m}{x_1,\dots,x_m} =
  \Jacobiandet{f_1,\dots,f_m}{x_1,\dots,x_m} \]
```

typesets as

$$\frac{\partial(f_1, \dots, f_m)}{\partial(x_1, \dots, x_m)} = \begin{vmatrix} \left(\frac{\partial f_1}{\partial x_1}\right)_{x_{j \neq 1}} & \dots & \left(\frac{\partial f_1}{\partial x_m}\right)_{x_{j \neq m}} \\ \vdots & & \vdots \\ \left(\frac{\partial f_m}{\partial x_1}\right)_{x_{j \neq 1}} & \dots & \left(\frac{\partial f_m}{\partial x_m}\right)_{x_{j \neq m}} \end{vmatrix}.$$

If the option `moles-range` or the `thermomolesrange` environment is used, the same code produces

$$\frac{\partial(f_1, \dots, f_m)}{\partial(x_1, \dots, x_m)} = \begin{vmatrix} \left(\frac{\partial f_1}{\partial x_1}\right)_{x_2, \dots, x_m} & \dots & \left(\frac{\partial f_1}{\partial x_m}\right)_{x_1, \dots, x_{m-1}} \\ \vdots & & \vdots \\ \left(\frac{\partial f_m}{\partial x_1}\right)_{x_2, \dots, x_m} & \dots & \left(\frac{\partial f_m}{\partial x_m}\right)_{x_1, \dots, x_{m-1}} \end{vmatrix}.$$

2.9 Sums and Products

`\sumall` It is common to require sums and products such as
`\sumallbutlast`
`\prodall`

$$\sum_{i=1}^C x_i = 1 \quad \text{or} \quad x_C = 1 - \sum_{i=1}^{C-1} x_i \quad \text{or} \quad \underline{G} = \sum_{j=1}^C \mu_j n_j \quad \text{and} \quad K = \prod_{k=1}^C a_k^{v_k}.$$

This package defines shortcuts to typeset such terms thus:

```
\[ \sumall_i x_i = 1 \quad \text{\text{or}} \quad \quad
  x_{\ncomponents} = 1 - \sumallbutlast_i x_i \quad \text{\text{or}} \quad \quad
  \underline{G} = \sumall_j \mu_j Nt_j \quad \text{\text{and}} \quad \quad
  K = \prodall_k a_k^{\nu_k} \quad \]
```

The symbol C can be changed by redefining the expandable macro `\ncomponents`. This is done automatically by some package options (e.g., `TesterModell` changes it to n ; `Sandler` changes it to C ; `Thompson` changes it to 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>
EUAGHan	\underline{E}	\underline{U}	\underline{A}	\underline{G}	\underline{H}	\underline{a}	n
EUAGHaN	\underline{E}	\underline{U}	\underline{A}	\underline{G}	\underline{H}	\underline{a}	N
EUHAGan	(synonym for EUAGHan)						
EUHAGaN	(synonym for EUAGHaN)						
EUFGHan	\underline{E}	\underline{U}	\underline{F}	\underline{G}	\underline{H}	\underline{A}	n
EUFGHAN	\underline{E}	\underline{U}	\underline{F}	\underline{G}	\underline{H}	\underline{A}	N
EEFGHan	$\underline{\mathcal{E}}$	\underline{E}	\underline{F}	\underline{G}	\underline{H}	\underline{A}	n
EEFGHAN	$\underline{\mathcal{E}}$	\underline{E}	\underline{F}	\underline{G}	\underline{H}	\underline{A}	N
EEFGHan	$\underline{\mathcal{E}}$	\underline{E}	\underline{F}	\underline{G}	\underline{H}	\underline{a}	n
EEFGHaN	$\underline{\mathcal{E}}$	\underline{E}	\underline{F}	\underline{G}	\underline{H}	\underline{a}	N
EEAGHan	$\underline{\mathcal{E}}$	\underline{E}	\underline{A}	\underline{G}	\underline{H}	\underline{a}	N
EEAGHAN	$\underline{\mathcal{E}}$	\underline{E}	\underline{A}	\underline{G}	\underline{H}	$\underline{\mathcal{A}}$	n
EUAGHAN	\underline{E}	\underline{U}	\underline{A}	\underline{G}	\underline{H}	$\underline{\mathcal{A}}$	N
EUFGHan	\underline{E}	\underline{U}	\underline{F}	\underline{G}	\underline{H}	\underline{a}	n
EUFGHAN	\underline{E}	\underline{U}	\underline{F}	\underline{G}	\underline{H}	\underline{a}	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; the manner of denoting extensive, molar, and specific properties; and the manner of writing the number of moles of all or most species in partial derivatives. 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:

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

Table 4. Notation sets that can be set using the options `intensive-plain` (the default), `extensive-plain`, `extensive-superscript`, and `intensive-lowercase`.

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

which would look like

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

Using the `EEFGHAN` option, the same markup would yield

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

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

```
\[ \Hpm_i = \Partial*\{\Ht\}\{\Nt_i\}\{T,P,\allNsbut{i}\}
= \Partial{\Nt\Hm}\{\Nt_i\}\{T,P,\allNsbut{i}\} \]
```

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.

thermoextensiveplain (*env.*) It is possible to change notation locally, though there are very, very few good reasons why you would want to do this in a regular document—normally, one would use the corresponding package options. The environment thermoextensiveplain (env.) means `\St` will become \underline{S} and `\Sm` will become S in the text. Similarly, inside thermoextensiveplain, `\St` will become S and `\Sm` will become \underline{S} . Inside thermoextensivesuperscript (env.) thermoextensivesuperscript, `\St` will become S^t and `\Sm` will be S ; and inside thermoextensivesubscript, `\St` will be S and `\Sm` will be s .

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:

```
\[ \Partial*\{Vm\}{T}{P} = \PartialMixSecond{Gm}{T}{P}{}
= \PartialMixSecond{Gm}{P}{T}{}
= -\Partial{Sm}{P}{T}. \]
```

Using the parentheses option (the default), this gives

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

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

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

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

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

Combined with plain-derivatives, this would give

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

The variables are sorted into an order defined by an internal constant, meaning T will always be listed before P . The order by default is in the order that terms appear in the fundamental equations, that is,

$$\begin{aligned} d\underline{U} &= T d\underline{S} - P d\underline{V} + \mu dn \\ d\underline{H} &= T d\underline{S} + \underline{V} dP + \mu dn \\ d\underline{A} &= -\underline{S} dT - P d\underline{V} + \mu dn \\ &\vdots \\ d\underline{\Omega} &= -\underline{S} dT - P d\underline{V} - n d\mu, \end{aligned}$$

with the exception that subscripted variables are (currently) not sortable. If `\Nt_i` or some similar construct appears as a variable and `\allNsbut{i}` appears in the held-constant list, the package will assume that the argument list *should* contain all of the mole numbers. Symbols without subscripts that are not in the fundamental equation or one of its variants are sorted in alphabetical order.

- `thermoparentheses (env.)` If you want to use parentheses *locally*, even though your overall document uses
- `thermobrackets (env.)` another delimiter, the `thermoparentheses` environment will do that. Similarly,
- `thermobar (env.)` `thermobrackets` will temporarily switch to brackets, `thermobar` will temporarily
- `thermoplain (env.)` switch to a tailing vertical bar, and `thermoplain` will remove delimiters altogether.
- `thermosubscripts (env.)` The environments `thermosubscripts` and `thermoN0subscripts` force the use or
- `thermoN0subscripts (env.)` disuse of subscripts, respectively.

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 `\allNsbut` 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.

- `thermomolesrange (env.)` The environment `thermomolesrange` temporarily redefines `\allNs` and other range-oriented macros as though the `moles-range` package option had been invoked.
- `\ncomponents` You can change the symbol for the number of components (default: C) by redefining the macro `\ncomponents`. This is done for you by some package options that define notation for particular textbooks.

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.

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 to another symbol if you want to switch to an option that redefines `\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 do not handle this case this way.

It should be noted that the `\dbar` macro is context-dependent: 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 Particular 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:

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 notation for specific and intensive properties and redefines `\cV`, `\cVs`, `\cVt`, `\expansivitysymbol`, and `\ncomponents` to match his notation.

CBK Notation used by Çengel, Boles, and Kanoğlu, *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 the internal symbol for pressure, `\Deltarxn`, `\compressibilitysymbol`, and `\expansivitysymbol` to fit their usage. Their prodigal symbols for specific and total volume, which appear to be from the font ITC Benguiat Gothic Standard Book Oblique, 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 `\IG`, `\IGM`, `\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`, `\fmix`, and `\phimix`, as well as symbols for the Helmholtz free energy and pressure, intensive and specific property notation, and partial molar notation to fit their somewhat ill-advised notation.

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`, `\Henryrat`, `\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`, `\Henryrat`, `\Henrymol`, `\allcomponents`, `\ncomponents`, `\IS`, `\IG`, `\IGM`, `\fpure`, `\hipure`,

`\mix`, `\Deltamix`, `\Deltarxn`, and `\allbut` to match his notation, plus adjustments to intensive (molar) properties. Note that he uses c , 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

We set up some non-standard token comparison variants; these are designed to catch both `\ncomponents=\ncomponents` and `\ncomponents=C` (using the default options); we have to define `\ncomponents` to be expandable to make these work at all.

```
1 \ExplSyntaxOn
2 \cs_generate_variant:Nn \tl_if_eq:nnTF { xxTF }
```

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.

```
3 \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*`, `\providecommand*`, `\NewDocumentCommand`, or `\ProvideDocumentCommand`) in the preamble will override the ones here.

```
4 \AtBeginDocument{
5   \@ifpackageloaded{pxfonts}{%
6     \ProvideDocumentCommand{\dbar}{}
7       {\mkern5mu\mathchar'26\mkern-10mu d}
8   }{}
9   \@ifpackageloaded{newpxmath}{%
10    \ProvideDocumentCommand{\dbar}{}
11      {\mkern5mu\mathchar'26\mkern-10mu d}
12  }{}
13  \@ifpackageloaded{txfonts}{%
14    \ProvideDocumentCommand{\dbar}{}
15      {\mkern5mu\mathchar'26\mkern-11mu d}
16  }{}
17  \@ifpackageloaded{mathptmx}{%
18    \ProvideDocumentCommand{\dbar}{}
19      {\mkern5mu\mathchar'26\mkern-10mu d}
20  }{}
}
```

```

21 \ifpackageloaded{newtxmath}{%
22   \ProvideDocumentCommand{\dbar}{}
23     {\mkern5mu\mathchar'26\mkern-11mu d}
24   }{}
25 \ifpackageloaded{mathdesign}{%
26   \tl_const:Nn \c_@@_charter_tl {mdbch}
27   \tl_const:Nn \c_@@_utopia_tl {mdput}
28   \tl_const:Nn \c_@@_garamond_tl {mdugm}
29   \tl_if_eq:NNT \MD@default@family \c_@@_utopia_tl
30   { \ProvideDocumentCommand{\dbar}{}
31     {\mkern5mu\mathchar'26\mkern-20mu d}
32   }
33   \tl_if_eq:NNT \MD@default@family \c_@@_charter_tl
34   { \ProvideDocumentCommand{\dbar}{}
35     {\mkern5mu\mathchar'26\mkern-15mu d}
36   }
37   \tl_if_eq:NNT \MD@default@family \c_@@_garamond_tl
38   { \ProvideDocumentCommand{\dbar}{}
39     {\mkern5mu\mathchar'26\mkern-17mu d}
40   }
41   }{}
42 % Defaults to Computer Modern
43 \ProvideDocumentCommand{\dbar}{}
44   {\mkern3mu\mathchar'26\mkern-12mu d}
45 }

```

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).

The default symbols are not intended to be easy to change—the intended mechanism is through package options. If you want to use a non-standard symbol that is not available through one of the package options, you can redefine the internal token lists inside `\ExplSyntaxOn... \ExplSyntaxOff`. For example,

```

\ExplSyntaxOn
  \tl_gset:Nn \g__thermodynamics_Helmholtz_symbol {H}
  \tl_gset:Nn \g__thermodynamics_enthalpy_symbol {h}
\ExplSyntaxOff

```

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, expansivites, 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.

```

46 \tl_new:N \g_@@_total_energy_symbol
47 \tl_new:N \g_@@_internal_energy_symbol
48 \tl_new:N \g_@@_Helmholtz_symbol
49 \tl_new:N \g_@@_Gibbs_symbol
50 \tl_new:N \g_@@_Landau_symbol
51 \tl_new:N \g_@@_enthalpy_symbol
52 \tl_new:N \g_@@_entropy_symbol
53 \tl_new:N \g_@@_area_symbol
54 \tl_new:N \g_@@_volume_symbol

```

```

55 \tl_new:N \g_@@_mole_symbol
56 \tl_new:N \g_@@_heat_symbol
57 \tl_new:N \g_@@_work_symbol
58 \tl_new:N \g_@@_temperature_symbol
59 \tl_new:N \g_@@_pressure_symbol
60
61 \tl_gset:Nn \g_@@_total_energy_symbol E
62 \tl_gset:Nn \g_@@_internal_energy_symbol U
63 \tl_gset:Nn \g_@@_Helmholtz_symbol A
64 \tl_gset:Nn \g_@@_Gibbs_symbol G
65 \tl_gset:Nn \g_@@_Landau_symbol \Omega
66 \tl_gset:Nn \g_@@_enthalpy_symbol H
67 \tl_gset:Nn \g_@@_entropy_symbol S
68 \tl_gset:Nn \g_@@_area_symbol a
69 \tl_gset:Nn \g_@@_volume_symbol V
70 \tl_gset:Nn \g_@@_mole_symbol n
71 \tl_gset:Nn \g_@@_heat_symbol Q
72 \tl_gset:Nn \g_@@_work_symbol W
73 \tl_gset:Nn \g_@@_temperature_symbol T
74 \tl_gset:Nn \g_@@_pressure_symbol P

```

We then define two functions and several lengths that we shall use when drawing rules 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.

```

75 \cs_new:Nn \@@_underline:n
76 { \mkern1mu\underline{\mkern-1mu #1\mkern-4mu}\mkern4mu }
77 \cs_new:Nn \@@_overline:n
78 { \mkern2mu\overline{\mkern-2mu #1\mkern-1mu}\mkern1mu }

```

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

```

79 \tl_new:N \l_@@_PartialOpen_tl
80 \tl_new:N \l_@@_PartialEmptyClose_tl
81 \tl_new:N \l_@@_PartialClose_tl
82
83 \tl_set:Nn \l_@@_PartialOpen_tl { ( }
84 \tl_set:Nn \l_@@_PartialClose_tl { ) }
85 \tl_set:Nn \l_@@_PartialEmptyClose_tl { }

```

4.2 Package Options

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

```

86 \DeclareOption{EUAGHan}{ }% the default
87 \DeclareOption{EUAGHaN}{\tl_gset:Nn \g_@@_mole_symbol N}%
88 \DeclareOption{EUHAGan}{\ExecuteOptions{EUAGHan}}

```

```

89 \DeclareOption{EUHAGaN}{\ExecuteOptions{EUAGHaN}}
90 \DeclareOption{EUFGHaN}{%
91   \tl_gset:Nn \g_@@_Helmholtz_symbol F
92   \tl_gset:Nn \g_@@_area_symbol A
93 }
94 \DeclareOption{EUFGHAN}{%
95   \tl_gset:Nn \g_@@_Helmholtz_symbol F
96   \tl_gset:Nn \g_@@_area_symbol A
97   \tl_gset:Nn \g_@@_mole_symbol N
98 }
99 \DeclareOption{EEFGHaN}{%
100  \tl_gset:Nn \g_@@_total_energy_symbol {\mathcal{E}}
101  \tl_gset:Nn \g_@@_internal_energy_symbol E
102  \tl_gset:Nn \g_@@_Helmholtz_symbol F
103  \tl_gset:Nn \g_@@_area_symbol A
104 }
105 \DeclareOption{EEFGHAN}{%
106  \tl_gset:Nn \g_@@_total_energy_symbol {\mathcal{E}}
107  \tl_gset:Nn \g_@@_internal_energy_symbol E
108  \tl_gset:Nn \g_@@_Helmholtz_symbol F
109  \tl_gset:Nn \g_@@_area_symbol A
110  \tl_gset:Nn \g_@@_mole_symbol N
111 }
112 \DeclareOption{EEFGHaN}{%
113  \tl_gset:Nn \g_@@_total_energy_symbol {\mathcal{E}}
114  \tl_gset:Nn \g_@@_internal_energy_symbol E
115  \tl_gset:Nn \g_@@_Helmholtz_symbol F
116 }
117 \DeclareOption{EEFGHaN}{%
118  \tl_gset:Nn \g_@@_total_energy_symbol {\mathcal{E}}
119  \tl_gset:Nn \g_@@_internal_energy_symbol E
120  \tl_gset:Nn \g_@@_Helmholtz_symbol F
121  \tl_gset:Nn \g_@@_mole_symbol N
122 }
123 \DeclareOption{EEAGHaN}{%
124  \tl_gset:Nn \g_@@_total_energy_symbol {\mathcal{E}}
125  \tl_gset:Nn \g_@@_internal_energy_symbol E
126 }
127 \DeclareOption{EEAGHaN}{%
128  \tl_gset:Nn \g_@@_total_energy_symbol {\mathcal{E}}
129  \tl_gset:Nn \g_@@_internal_energy_symbol E
130  \tl_gset:Nn \g_@@_mole_symbol N
131 }
132 \DeclareOption{EUAGHaN}{%
133  \tl_gset:Nn \g_@@_area_symbol {\mathcal{A}}
134 }
135 \DeclareOption{EUAGHAN}{%
136  \tl_gset:Nn \g_@@_area_symbol {\mathcal{A}}
137  \tl_gset:Nn \g_@@_mole_symbol N
138 }
139 \DeclareOption{EUFGHaN}{%
140  \tl_gset:Nn \g_@@_Helmholtz_symbol F
141 }
142 \DeclareOption{EUFGHaN}{%
143  \tl_gset:Nn \g_@@_Helmholtz_symbol F
144  \tl_gset:Nn \g_@@_mole_symbol N

```


145 }

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.

```
146 \DeclareOption{delta}{ \cs_set_eq:NN \dbar \delta }
```

Next, we define options for the set of notation. The default is `intensive-plain`, which produces things like V for molar volume, \underline{V} for total volume, and \hat{V} for specific volume. The symbols themselves are produced via internal (non-user-facing) macros.

```
147 \cs_new:Nn \@@_extensive:n {#1}
148 \cs_new:Nn \@@_intensive:n {#1}
149 \cs_new:Nn \@@_specific:n {\hat{#1}}
150
151 \cs_new:Npn \@@_set_intensive_plain
152 {
153   \cs_set:Nn \@@_extensive:n {\@@_underline:n{##1}}
154   \cs_set:Nn \@@_intensive:n {##1}
155 }
156 \cs_new:Npn \@@_set_extensive_plain
157 {
158   \cs_set:Nn \@@_extensive:n {##1}
159   \cs_set:Nn \@@_intensive:n {\@@_underline:n{##1}}
160 }
161 \cs_new:Npn \@@_set_lowercase_pms
162 {
163   \RenewDocumentCommand{\partialmolar}{m}
164   {
165     \tl_set:Nn \l_@@_pm_symbol_tl {\text_lowercase:n {##1}}
166     \@@_generic_pm:
167   }
168 }
169 \cs_new:Npn \@@_set_intensive_lowercase
170 {
171   \cs_set:Nn \@@_extensive:n {\text_uppercase:n {##1}}
172   \cs_set:Nn \@@_intensive:n {\text_lowercase:n {##1}}
173   \cs_set:Nn \@@_specific:n {\hat{\text_lowercase:n {##1}}}
174 }
175 \cs_new:Npn \@@_set_extensive_superscripts
176 {
177   \cs_set:Nn \@@_extensive:n {##1\c_math_superscript_token t}
178   \cs_set:Nn \@@_intensive:n {##1}
179 }
180 \DeclareOption{extensive-plain}{\@@_set_extensive_plain}
181 \DeclareOption{intensive-plain}{\@@_set_intensive_plain} % the default
182 \DeclareOption{intensive-lowercase}{% PLEASE don't use this!
183   \@@_set_intensive_lowercase
184   \AtEndOfPackage{
185     \@@_set_lowercase_pms
186     \RenewDocumentCommand{\heatcapacitysymbol}{}{c}
187   }
188 }
189 \DeclareOption{extensive-superscript}{%
190   \@@_set_extensive_superscripts
191   \AtEndOfPackage{%
192     \RenewDocumentCommand{\URt}{}{\g_@@_internal_energy_symbol
193       \c_math_superscript_token{residual,t}}
```

```

194 \RenewDocumentCommand{\HRt}{}{\g_@@_enthalpy_symbol
195   \c_math_superscript_token{residual,t}}
196 \RenewDocumentCommand{\FRt}{}{\g_@@_Helmholtz_symbol
197   \c_math_superscript_token{residual,t}}
198 \RenewDocumentCommand{\GRt}{}{\g_@@_Gibbs_symbol
199   \c_math_superscript_token{residual,t}}
200 \RenewDocumentCommand{\VRt}{}{\g_@@_volume_symbol
201   \c_math_superscript_token{residual,t}}
202 \RenewDocumentCommand{\SRt}{}{\g_@@_entropy_symbol
203   \c_math_superscript_token{residual,t}}
204 \RenewDocumentCommand{\Uet}{}{\g_@@_internal_energy_symbol
205   \c_math_superscript_token{excess,t}}
206 \RenewDocumentCommand{\HET}{}{\g_@@_enthalpy_symbol
207   \c_math_superscript_token{excess,t}}
208 \RenewDocumentCommand{\Fet}{}{\g_@@_Helmholtz_symbol
209   \c_math_superscript_token{excess,t}}
210 \RenewDocumentCommand{\GET}{}{\g_@@_Gibbs_symbol
211   \c_math_superscript_token{excess,t}}
212 \RenewDocumentCommand{\Vet}{}{\g_@@_volume_symbol
213   \c_math_superscript_token{excess,t}}
214 \RenewDocumentCommand{\SET}{}{\g_@@_entropy_symbol
215   \c_math_superscript_token{excess,t}}
216 }
217 }

```

The next two options choose whether variables held constant are subscripted (the default) or placed next to the function. 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.

```

218 \bool_new:N \l_@@_subscripted_bool
219 \bool_set_true:N \l_@@_subscripted_bool
220 \DeclareOption{subscripts}{\bool_set_true:N \l_@@_subscripted_bool}
221 \DeclareOption{nosubscripts}{\bool_set_false:N \l_@@_subscripted_bool}

```

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.

```

222 \DeclareOption{parentheses}{}
223 \DeclareOption{brackets}{%
224   \tl_set:Nn \l_@@_PartialOpen_tl {[ ]}
225   \tl_set:Nn \l_@@_PartialClose_tl { ] }
226   \tl_set:Nn \l_@@_PartialEmptyClose_tl { ] }
227 }
228 \DeclareOption{braces}{%
229   \tl_set:Nn \l_@@_PartialOpen_tl {\{ }
230   \tl_set:Nn \l_@@_PartialClose_tl {\} }
231   \tl_set:Nn \l_@@_PartialEmptyClose_tl {\} }
232 }
233 \DeclareOption{bar}{%
234   \tl_set:Nn \l_@@_PartialOpen_tl { . }
235   \tl_set:Nn \l_@@_PartialClose_tl {\rvert }
236   \tl_set:Nn \l_@@_PartialEmptyClose_tl { . }
237 }
238 \DeclareOption{plain-derivatives}{% This implies dU(S,V,N)/dS notation

```

```

239 \tl_set:Nn \l_@@_PartialOpen_tl {.}
240 \tl_set:Nn \l_@@_PartialClose_tl {.}
241 \tl_set:Nn \l_@@_PartialEmptyClose_tl {.}
242 \ExecuteOptions{nosubscripts}
243 }

```

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. The command is expandable so we can perform comparisons to user-entered values.

```
244 \NewExpandableDocumentCommand \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`, `\allYs` `\allMsbut`) and chemical potentials (`\allmus`, `\allmusbut`), which occur frequently in mixture thermodynamics. The default is for `\allNs` to become \bar{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.³ Essentially identical commands are defined for chemical potentials and masses: `\allmus` and `\allmusbut` and `\allMs` and `\allMsbut`, respectively.

```

245 \NewDocumentCommand{\allNs}{0{i}}{\allcomponents[#1]{\Nt}}
246 \NewDocumentCommand{\allXs}{0{i}}{\allcomponents[#1]{x}}
247 \NewDocumentCommand{\allYs}{0{i}}{\allcomponents[#1]{y}}
248 \NewDocumentCommand{\allmus}{0{i}}{\allcomponents[#1]{\mu}}
249 \NewDocumentCommand{\allMs}{0{i}}{\allcomponents[#1]{m}}
250 \NewDocumentCommand{\allWs}{0{i}}{\allcomponents[#1]{w}}

```

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

$$\left[\frac{\partial \{G_m\}_T}{\partial \{x_i\}_T, P, \{x_{j \neq i}\}} = -\sum \quad \neq \bar{G}_i \right]$$

yields

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

```

251 \NewDocumentCommand{\allNsbut}{0{j} m} {\allbut[#1]{#2}{\Nt}}
252 \NewDocumentCommand{\allXsbut}{0{j} m} {\allbutlastand[#1]{#2}{x}}
253 \NewDocumentCommand{\allYsbut}{0{j} m} {\allbutlastand[#1]{#2}{y}}
254 \NewDocumentCommand{\allmusbut}{0{j} m} {\allbut[#1]{#2}{\mu}}
255 \NewDocumentCommand{\allMsbut}{0{j} m} {\allbut[#1]{#2}{m}}
256 \NewDocumentCommand{\allWsbut}{0{j} m} {\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

```
\allcomponents \NewDocumentCommand{\allZsbut}{0{j} m}{\allbutlastand[#1]{#2}{z}}
```

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

```
\NewDocumentCommand{\allCs}{0{}}{\allcomponents{C}}
```

³The index j is automatically replaced with k if the user issues `\allNsbut{j}`.

```

257 \NewDocumentCommand{\allcomponents}{0{} m}{\vec{#2}}
258 \NewDocumentCommand{\allbut}{0{j} m m}
259 { \tl_if_eq:nnTF {#1} {#2}
260   { {#3}\c_math_subscript_token{k \neq #2} }
261   { {#3}\c_math_subscript_token{#1 \neq #2} }
262 }
263 \NewDocumentCommand{\allbutlastand}{0{j} m m}
264 { \tl_if_eq:xxTF {#2} {\ncomponents}
265   { {#3}\c_math_subscript_token{#1 \neq #2} }
266   { \tl_if_eq:nnTF {#1} {#2}
267     { {#3}\c_math_subscript_token{k \neq #2,\ncomponents} }
268     { {#3}\c_math_subscript_token{#1 \neq #2,\ncomponents} }
269   }
270 }

```

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

```

271 \DeclareOption{moles-index}{}
272 \DeclareOption{moles-range}{ \@@_set_moles_range }
273 \cs_new:Npn \@@_set_moles_range {%
274   \RenewDocumentCommand{\allcomponents}{0{} m}
275   { {##2}\c_math_subscript_token 1,\dots,
276     {##2}\c_math_subscript_token{\ncomponents} }
277   \RenewDocumentCommand{\allbut}{0{j} m m}
278   { \tl_if_eq:nnTF {##2} {1}
279     { {##3}\c_math_subscript_token 2,\dots,
280       {##3}\c_math_subscript_token{\ncomponents} }
281     { \tl_if_eq:xxTF {##2} {\ncomponents}
282       { {##3}\c_math_subscript_token 1,\dots,
283         {##3}\c_math_subscript_token{\ncomponents-1} }
284       { {##3}\c_math_subscript_token 1,\dots,
285         [{##3}\c_math_subscript_token{##2}],
286         \dots,{##3}\c_math_subscript_token{\ncomponents} } }
287   }
288 }
289 \RenewDocumentCommand{\allbutlastand}{0{j} m m}
290 { \tl_if_eq:nnTF {##2} {1}
291   { {##3}\c_math_subscript_token 2,\dots,
292     {##3}\c_math_subscript_token{\ncomponents-1} }
293   { \tl_if_eq:xxTF {##2} {\ncomponents}
294     { {##3}\c_math_subscript_token 1,\dots,
295       {##3}\c_math_subscript_token{\ncomponents-1}
296     }
297     { \tl_if_eq:xxTF {##2} {\ncomponents-1}
298       { {##3}\c_math_subscript_token 1,\dots,
299         {##3}\c_math_subscript_token{\ncomponents-2} }
300       {
301         {##3}\c_math_subscript_token 1,\dots,
302         [{##3}\c_math_subscript_token{##2}],\dots,
303         {##3}\c_math_subscript_token{\ncomponents-1}
304       }
305     }
306   }
307 }
308 }

```

The remaining options define textbook-specific notation.

```

309 \DeclareOption{Bejan}{

```

```

310 \ExecuteOptions{EUFGHAN,intensive-lowercase,delta}
311 \cs_set:Nn \@@_specific:n {\text_lowercase:n {#1}}
312 \cs_set:Nn \@@_intensive:n {\bar{\text_lowercase:n {#1}}}
313 \tl_gset:Nn \g_@@_volume_symbol v
314 }
315 \DeclareOption{CBK}{
316 \ExecuteOptions{EUAGHAN,intensive-lowercase}
317 \AtEndOfPackage{
318 \cs_set:Nn \@@_overline:n {\widetilde{#1}}
319 \cs_set:Nn \@@_specific:n {\text_lowercase:n {#1}}
320 \cs_set:Nn \@@_intensive:n {\bar{\text_lowercase:n {#1}}}
321 \tl_gset:Nn \g_@@_pressure_symbol p
322 \RenewDocumentCommand{\Deltarxn}{m}{#1}\c_math_subscript_token R}
323 \RenewDocumentCommand{\compressibilitysymbol}{}{\beta}
324 \RenewDocumentCommand{\expansivitysymbol}{}{\alpha}
325 }
326 }
327 \DeclareOption{ElliottLira}{
328 \AtEndOfPackage{
329 \RenewDocumentCommand{\allcomponents}{0}{m}{#2}
330 \RenewDocumentCommand{\Deltarxn}{m}{\Delta #1}
331 \RenewDocumentCommand{\IG}{}{\text{ig}}
332 \RenewDocumentCommand{\IGM}{}{\text{ig}}
333 \RenewDocumentCommand{\IS}{}{\text{is}}
334 }
335 }
336 \DeclareOption{Koretsky}{
337 \ExecuteOptions{EUAGHAN,brackets,intensive-lowercase,delta}
338 \AtEndOfPackage{
339 %^^A Undo part of intensive-lowercase
340 \RenewDocumentCommand{\partialmolar}{m}
341 { \tl_set:Nn \l_@@_pm_symbol_tl {#1}
342 \@@_generic_pm:
343 }
344 \RenewDocumentCommand{\expansivitysymbol}{}{\beta}
345 \RenewDocumentCommand{\IS}{}{\text{ideal}}
346 \RenewDocumentCommand{\residual}{}{\text{dep}}
347 \RenewDocumentCommand{\IG}{}{\text{ideal}}
348 \RenewDocumentCommand{\IGM}{}{\text{ideal}}
349 \RenewDocumentCommand{\Henryrat}{}{\mathcal{H}}
350 \RenewDocumentCommand{\gammarat}{}
351 {\gamma\c_math_superscript_token\text{Henry's}}
352 \RenewDocumentCommand{\phipure}{}{\varphi}
353 \RenewDocumentCommand{\phimix}{}{\hat{\varphi}}
354 \RenewDocumentCommand{\phisat}{}{\varphi\c_math_superscript_token\sat}
355 \cs_new:Npn \Delta_fus_sym {} {}
356 \NewSubscriptedSymbol{\Delta_fus}{\Delta_fus_sym}{\fusion}
357 \RenewDocumentCommand{\Deltafus}{m}{
358 \cs_set:Npn \Delta_fus_sym {} { \Delta #1 }
359 \Delta_fus
360 }
361 \cs_new:Npn \Delta_vap_sym {} {}
362 \NewSubscriptedSymbol{\Delta_vap}{\Delta_vap_sym}{\vaporization}
363 \RenewDocumentCommand{\Deltavap}{m}{
364 \cs_set:Npn \Delta_vap_sym {} { \Delta #1 }
365 \Delta_vap

```

```

366 }
367 \cs_new:Npn \Delta_sub_sym {} {}
368 \NewSubscriptedSymbol{\Delta_sub}{\Delta_sub_sym}{\sublimation}
369 %^^A Undo part of intensive-lowercase
370 \RenewDocumentCommand{\Deltasub}{m}{
371   \cs_set:Npn \Delta_sub_sym {} { \Delta #1 }
372   \Delta_sub
373 }
374 }
375 }
376 \DeclareOption{MSBB}{
377   \ExecuteOptions{EUFGHAn,intensive-lowercase,delta}
378   \AtEndOfPackage{
379     \RenewDocumentCommand{\IGM}{}{\ast}
380     \RenewDocumentCommand{\IG}{}{\ast}
381     \RenewDocumentCommand{\expansivitysymbol}{}{\beta}
382     \RenewDocumentCommand{\allcomponents}{O{} m}{#2}
383     \RenewDocumentCommand{\allbut}{O{j} m m}
384     {
385       \tl_if_eq:nnTF {#1} {#2}
386       { {#3}\c_math_subscript_token k }
387       { {#3}\c_math_subscript_token{#1} }
388     }
389     \tl_gset_eq:NN \g_@@_Helmholtz_symbol \psi
390     \RenewDocumentCommand{\Ft}{}{\Psi}
391     \cs_set:Nn \@@_intensive:n {\@@_overline:n{\text_lowercase:n{#1}}}
392     \cs_set:Nn \@@_specific:n {\text_lowercase:n{#1}}
393     \RenewDocumentCommand{\fmix}{}{\bar f}
394     \RenewDocumentCommand{\phimix}{}{\bar\phi}
395     \RenewDocumentCommand{\phimix}{}{\bar\phi}
396     \tl_gset:NN \g_@@_pressure_symbol p
397     \RenewDocumentCommand{\partialmolar}{m}
398     {
399       \tl_set:Nn \l_@@_pm_symbol_tl {#1}
400       \@@_generic_pm:
401     }
402   }
403 }
404 \DeclareOption{Prausnitz}{
405   \ExecuteOptions{intensive-lowercase}
406   \AtEndOfPackage{
407     \RenewDocumentCommand{\fmix}{}{f}
408     \RenewDocumentCommand{\phimix}{}{\phi}
409     \RenewDocumentCommand{\fsat}{}{\fpure\c_math_superscript_token\sat}
410     % TODO: this should pick up H_2 and make it into H_{2,1} (assuming the
411     % solvent is always 1...?)
412     \RenewDocumentCommand{\Henryrat}{}{H}
413     \RenewDocumentCommand{\residual}{}{\mathcal{R}}
414     \RenewDocumentCommand{\allcomponents}{O{i} m}
415     { {#2}\c_math_subscript_token{#1} }
416     \RenewDocumentCommand{\allbut}{O{i} m m}
417     { \tl_if_eq:nnTF {#1} {#2}
418       { {#3}\c_math_subscript_token k }
419       { {#3}\c_math_subscript_token{#1} }
420     }
421     \RenewSubscriptedSymbol{\fpure}{f}{\text{pure}}

```

```

422   \RenewSubscriptedSymbol{\phipure}{\phi}{\text{pure}}
423 }
424 }
425 \DeclareOption{Sandler}{
426   \ExecuteOptions{EUAGHaN,extensive-plain}
427   \AtEndOfPackage{
428     \RenewDocumentCommand{\sat}{}{\text{vap}}
429     \RenewDocumentCommand{\excess}{}{\text{ex}}
430     \RenewDocumentCommand{\residual}{}{\text{r}}
431     \RenewExpandableDocumentCommand{\ncomponents}{}{\mathcal{C}}
432     \RenewDocumentCommand{\fmix}{}{\bar f}
433     \RenewDocumentCommand{\fstd}{}{\bar f}_{c\_math\_superscript\_token\std}
434     \RenewDocumentCommand{\phimix}{}{\bar\phi}
435     \RenewDocumentCommand{\allcomponents}{0}{m}{@@_underline:n{#2}}
436     \RenewDocumentCommand{\IG}{}{\text{IG}}
437     \RenewDocumentCommand{\IGM}{}{\text{IGM}}
438     \RenewDocumentCommand{\IS}{}{\text{IM}}
439     \RenewDocumentCommand{\Deltamix}{m}
440       {\Delta}_{c\_math\_subscript\_token\mixing #1}
441     \RenewDocumentCommand{\Deltarxn}{m}
442       {\Delta}_{c\_math\_subscript\_token\reaction #1}
443     \RenewDocumentCommand{\Deltasub}{m}
444       {\Delta}_{c\_math\_subscript\_token\sublimation #1}
445     \RenewDocumentCommand{\Deltafus}{m}
446       {\Delta}_{c\_math\_subscript\_token\fusion #1}
447     \RenewDocumentCommand{\Deltavap}{m}
448       {\Delta}_{c\_math\_subscript\_token\vaporization #1}
449     \RenewDocumentCommand{\Henryrat}{}{H}
450     \RenewSubscriptedSymbol{cV}
451       {\heatcapacitiesymbol}{g}_@@_volume\_symbol}
452     \RenewSubscriptedSymbol{cP}
453       {\heatcapacitiesymbol}{g}_@@_pressure\_symbol}
454     \RenewSubscriptedSymbol{cVt}
455       {\Nt\heatcapacitiesymbol}{g}_@@_volume\_symbol}
456     \RenewSubscriptedSymbol{cPt}
457       {\Nt\heatcapacitiesymbol}{g}_@@_pressure\_symbol}
458   }
459 }
460 \DeclareOption{SVNAS}{
461   \ExecuteOptions{extensive-superscript}
462   \AtEndOfPackage{
463     \RenewDocumentCommand{\allcomponents}{0}{m}{#2}
464     \RenewDocumentCommand{\allbut}{0}{j}{m}{
465       {
466         \tl_if_eq:nnTF {#1} {#2}
467         { {#3}\c\_math\_subscript\_token k }
468         { {#3}\c\_math\_subscript\_token{#1} }
469       }
470     \RenewDocumentCommand{\IG}{}{\ig}
471     \RenewDocumentCommand{\IGM}{}{\ig}
472     \RenewDocumentCommand{\IS}{}{\id}
473     \RenewDocumentCommand{\expansivitysymbol}{}{\beta}
474     \RenewDocumentCommand{\Deltarxn}{m}{\Delta #1}
475     \RenewSubscriptedSymbol{cVt}
476       {\Nt\heatcapacitiesymbol}{g}_@@_volume\_symbol}
477     \RenewSubscriptedSymbol{cPt}

```

```

478         {\Nt\heatcapacitysymbol}{\g_@@_pressure_symbol}
479     }
480 }
481 \DeclareOption{TesterModell}{
482     \ExecuteOptions{EUAGHaN,delta}
483     \AtEndOfPackage{
484         \RenewExpandableDocumentCommand{\ncomponents}{}{n}
485         \RenewDocumentCommand{\allcomponents}{0{i} m}
486         {
487             {#2}\c_math_subscript_token{#1}
488         }
489         \RenewDocumentCommand{\allbut}{0{i} m m}
490         { \tl_if_eq:nnTF {#1} {#2}
491             { {#3}\c_math_subscript_token{k}[#2] }
492             { {#3}\c_math_subscript_token{#1}[#2] }
493         }
494         \RenewDocumentCommand{\allbutlastand}{0{j} m m}
495         { \tl_if_eq:xxTF {#2} {\ncomponents}
496             { {#3}\c_math_subscript_token{#1}\relax[#2] }
497             { \tl_if_eq:nnTF {#1} {#2}
498                 { {#3}\c_math_subscript_token{k}[#2,\ncomponents] }
499                 { {#3}\c_math_subscript_token{#1}[#2,\ncomponents] }
500             }
501         }
502         \RenewDocumentCommand{\IG}{}{}{o}
503         \RenewDocumentCommand{\IGM}{}{}{o}
504         \RenewDocumentCommand{\IS}{}{}{ID}
505         \RenewDocumentCommand{\excess}{}{}{EX}
506         \RenewDocumentCommand{\reaction}{}{}{rx}
507         \RenewDocumentCommand{\Henryrat}{}
508             {f\c_math_superscript_token{\ast\ast}}
509         \RenewDocumentCommand{\Henrymol}{}{}{f\c_math_superscript_token\ast}
510         \RenewDocumentCommand{\gammarat}{}
511             {\gamma\c_math_superscript_token{\ast\ast}}
512         \RenewDocumentCommand{\gammamol}{}
513             {\gamma\c_math_superscript_token\ast}
514         \RenewExpandableDocumentCommand{\JTsymbol}{}{\alpha}
515         \RenewSubscriptedSymbol{\muJT}{\JTsymbol}{H}
516     }
517 }
518 \DeclareOption{Thompson}{
519     \ExecuteOptions{EUAGHaN,delta}
520     \AtEndOfPackage
521     {
522         \RenewDocumentCommand{\excess}{}{}{EX}
523         \RenewDocumentCommand{\residual}{}{}{R}
524         %^^A He uses k_H for both types of Henry's constant, but I changed
525         %^^A the rational basis one just so there is SOME difference
526         \RenewSubscriptedSymbol{\Henryrat}{k}{h}
527         \RenewSubscriptedSymbol{\Henrymol}{k}{H}
528         \RenewDocumentCommand{\allcomponents}{0{j} m}
529         {
530             {#2}\c_math_subscript_token{#1}
531         }
532         \RenewDocumentCommand{\allNs}{0{j}}{\allcomponents[#1]{\Nt}}
533         \RenewDocumentCommand{\allXs}{0{j}}{\allcomponents[#1]{x}}

```



```

534 \RenewDocumentCommand{\allYs}{0{j}}{\allcomponents[#1]{y}}
535 \RenewDocumentCommand{\allmus}{0{j}}{\allcomponents[#1]{\mu}}
536 \RenewDocumentCommand{\allMs}{0{j}}{\allcomponents[#1]{m}}
537 \RenewDocumentCommand{\allWs}{0{j}}{\allcomponents[#1]{w}}
538 \RenewExpandableDocumentCommand{\ncomponents}{}{c}
539 \RenewDocumentCommand{\IS}{}{IS}
540 \RenewDocumentCommand{\IG}{}{IG}
541 \RenewDocumentCommand{\IGM}{}{IG}
542 \cs_new:Nn \@@_fpure_one:n
543 {
544   f\c_math_subscript_token{#1}
545   \peek_catcode_remove:NF \c_math_superscript_token
546   { \c_math_superscript_token\bullet }
547 }
548 \RenewDocumentCommand{\fpure}{}
549 {
550   \peek_catcode_remove:NTF \c_math_subscript_token
551   { \@@_fpure_one:n }
552   { f }
553 }
554 \cs_new:Nn \@@_intensive_two:n
555 {
556   \c_math_subscript_token{#1}
557   \peek_catcode:NF \c_math_superscript_token
558   {
559     \c_math_superscript_token\bullet
560   }
561 }
562 \cs_set:Nn \@@_intensive:n
563 { #1
564   \peek_catcode_remove:NT \c_math_subscript_token
565   { \@@_intensive_two:n }
566 }
567 \cs_new:Nn \@@_phipure_one:n
568 {
569   \phi\c_math_subscript_token{#1}
570   \peek_catcode:NF \c_math_superscript_token
571   { \c_math_superscript_token\bullet }
572 }
573 \RenewDocumentCommand{\phipure}{}
574 {
575   \peek_catcode_remove:NTF \c_math_subscript_token
576   {
577     \@@_phipure_one:n
578   }
579   {\phi}
580 }
581 \RenewDocumentCommand{\mixing}{}{MIX}
582 \RenewDocumentCommand{\Deltamix}{m}
583   {\Delta\c_math_subscript_token\mixing #1}
584 \RenewDocumentCommand{\Deltarxn}{m}{\Delta #1}
585 \RenewDocumentCommand{\allbut}{0{j} m m}
586 { \tl_if_eq:nnTF {#1} {#2}
587   {
588     {#3}\c_math_subscript_token k\neq{#3}\c_math_subscript_token{#2}
589   }

```

```

590     {
591     {#3}\c_math_subscript_token{#1}\neq{#3}\c_math_subscript_token{#2}
592     }
593   }
594 }
595 }

```

We execute the default options below.

```

596 \ExecuteOptions{EUAGHan,subscripts,parentheses,intensive-plain,moles-index}
597 \ProcessOptions

```

We next encode a routine to sort non-subscripted variables into a consistent order. It currently does not sort variables with subscripts.

```

598 \tl_const:Nn \c_@@_sort_order_tl
599   {\Et\Em\Es\Ut\Um\Us\Ht\Hm\Hs\Ft\Fm\Fs\Gt\Gm\Gs\Lt\Lm\Ls T\St\Sm\Ss
600     P\Vt\Vm\Vs\mu\Nt mwxzy\At\Am\As\sigma
601     ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyzy}
602 \clist_new:N \l_@@_in_list_clist
603 \clist_new:N \l_@@_sorted_list_clist
604 \clist_new:N \l_@@_remaining_list_clist
605 \cs_new:Nn \@@_sort_clist:n
606 {% Sort the list in the order of \c_@@_sort_order_tl
607
608 % Wipe out any remnants from the last sort
609 \clist_clear:N \l_@@_in_list_clist
610 \clist_clear:N \l_@@_sorted_list_clist
611
612 % Make a copy of the list
613 \clist_set:Nn \l_@@_remaining_list_clist {#1}
614
615 % Make a list of everything that's in the known sort order list
616 % and put everything else in the "not in sort order list" list.
617 \tl_map_inline:Nn \c_@@_sort_order_tl
618 {
619   \clist_if_in:NnT \l_@@_remaining_list_clist {##1}
620   { \clist_put_right:Nn \l_@@_in_list_clist {##1} }
621
622   \clist_remove_all:Nn \l_@@_remaining_list_clist {##1}
623 }
624
625 % Then merge the lists back together again.
626 \clist_if_empty:NF \l_@@_in_list_clist
627 {
628   \clist_put_right:Nn \l_@@_sorted_list_clist \l_@@_in_list_clist
629 }
630 \clist_if_empty:NF \l_@@_remaining_list_clist
631 {
632   \clist_put_right:Nn \l_@@_sorted_list_clist \l_@@_remaining_list_clist
633 }
634 \clist_use:Nn \l_@@_sorted_list_clist ,
635 }

```

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.

```
636 \dim_new:N \l_@@_Partial_const_dim
637 \dim_new:N \l_@@_operator_width_dim
638 \dim_new:N \l_@@_adjust_width_dim
639 \settowidth{\l_@@_operator_width_dim}{=}
640 \dim_set:Nn \l_@@_adjust_width_dim {0.1\l_@@_operator_width_dim}
641 \dim_add:Nn \l_@@_operator_width_dim \l_@@_adjust_width_dim
```

`\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.

```
642 \tl_new:N \l_@@_Partial_start_tl
643 \tl_new:N \l_@@_Partial_end_tl
644 \tl_new:N \l_@@_Partial_empty_end_tl
645 \tl_new:N \l_@@_Partial_middle_tl
646 \tl_set:Nn \l_@@_Partial_start_tl {\left\l_@@_PartialOpen_tl}
647 \tl_set:Nn \l_@@_Partial_end_tl {\right\l_@@_PartialClose_tl}
648 \tl_set:Nn \l_@@_Partial_empty_end_tl {\right\l_@@_PartialEmptyClose_tl}
649 \tl_set:Nn \l_@@_Partial_middle_tl {\middle}
650 \cs_set_eq:NN \@@_frac:nn \frac
651 \NewDocumentCommand{\Partial}{s m m m}
652 { \bool_if:nTF {#1}
653   {% Starred form (recursive)
654     \settowidth{\l_@@_Partial_const_dim}{\ensuremath{#4}}%
655     \dim_add:Nn \l_@@_Partial_const_dim {-0.20\l_@@_Partial_const_dim}%
656     \Partial{#2}{#3}{#4}%
657     \bool_if:NT \l_@@_subscripted_bool
658     { \dim_compare:nNnTF \l_@@_operator_width_dim
659       < \l_@@_Partial_const_dim
660       { \kern -\l_@@_operator_width_dim }
661       { \kern -\l_@@_Partial_const_dim }
662     }
663   }
664   {% Unstarred form
665     \bool_if:NTF \l_@@_subscripted_bool
666     {% Handle case of empty variables held constant
667       \tl_if_eq:nnTF {#4} {}
668       { \l_@@_Partial_start_tl
669         \@@_frac:nn{\partial #2}{\partial #3}\l_@@_Partial_empty_end_tl
670       }
671       { \l_@@_Partial_start_tl\@@_frac:nn{\partial #2}
672         {\partial #3}\l_@@_Partial_end_tl
673         \c_math_subscript_token{#4}%
674       }
675     }
676     {% Check whether #4 contains \allNsbut{i} and #3 is \Nt_i
677       \tl_if_in:nnTF {#3} {\Nt}
678       { \RenewDocumentCommand{\allbut}{0{j} m m}{\allcomponents{##3}}
679         \l_@@_Partial_start_tl
```

```

680     \@@_frac:nn{\partial #2(\@@_sort_clist:n{#4})}
681         {\partial #3}\l_@@_Partial_end_tl
682     }
683     { \l_@@_Partial_start_tl
684     \@@_frac:nn{\partial #2(\@@_sort_clist:n{#3,#4})}
685         {\partial #3}\l_@@_Partial_end_tl
686     }
687 }
688 }
689 }

```

`\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.

```

690 \NewDocumentCommand{\PartialBigg}{}
691 { \tl_set:Nn \l_@@_Partial_start_tl {\Biggl\l_@@_PartialOpen_tl}
692   \tl_set:Nn \l_@@_Partial_end_tl {\Biggr\l_@@_PartialClose_tl}
693   \tl_set:Nn \l_@@_Partial_Empty_end_tl
694     {\Biggr\l_@@_PartialEmptyClose_tl}
695   \Partial
696 }

```

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

```

697 \NewDocumentCommand{\Partialbigg}{}
698 { \tl_set:Nn \l_@@_Partial_start_tl {\biggl\l_@@_PartialOpen_tl}
699   \tl_set:Nn \l_@@_Partial_end_tl {\biggr\l_@@_PartialClose_tl}
700   \tl_set:Nn \l_@@_Partial_empty_end_tl
701     {\biggr\l_@@_PartialEmptyClose_tl}
702   \Partial
703 }

```

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

```

704 \NewDocumentCommand{\PartialSecond}{s m m m}
705 {
706   \bool_if:nTF {#1}
707     {% Starred form
708       \settoheight{\l_@@_Partial_const_dim}{\ensuremath{#4}}%
709       \dim_add:Nn \l_@@_Partial_const_dim {-0.20\l_@@_Partial_const_dim}
710       \PartialSecond{#2}{#3}{#4}%
711       \bool_if:nT \l_@@_subscripted_bool
712       { \dim_compare:nNnTF {\l_@@_operator_width_dim}
713         < {\l_@@_Partial_const_dim}
714         { \kern -\l_@@_operator_width_dim }
715         { \kern -\l_@@_Partial_const_dim }
716       }
717     }
718     {% Unstarred form
719       \bool_if:NTF \l_@@_subscripted_bool
720       {% Handles case of empty variables held constant
721         \tl_if_eq:nnTF {#4} {}
722         { \l_@@_Partial_start_tl
723           \@@_frac:nn{\partial\c_math_superscript_token 2 #2}
724           {\partial #3\c_math_superscript_token 2}\l_@@_Partial_empty_end_tl
725         }
726         { \l_@@_Partial_start_tl

```

```

727     \@@_frac:nn{\partial\c_math_superscript_token 2 #2}
728     {\partial #3\c_math_superscript_token 2}\l_@@_Partial_end_tl
729     \c_math_subscript_token{#4}%
730   }
731 }
732 {% Check whether #4 contains \allNsbut{i} and #3 is \Nt_i
733 \tl_if_in:nnTF {#2} {\Nt}
734 { \RenewDocumentCommand{\allbut}{0{j} m m}{\allcomponents{##3}}
735 \l_@@_Partial_start_tl
736 \@@_frac:nn{\partial\c_math_superscript_token 2 #2(#4)}
737 {\partial\c_math_superscript_token 2 #3}\l_@@_Partial_end_tl
738 }
739 { \l_@@_Partial_start_tl
740 \@@_frac:nn{\partial\c_math_superscript_token 2
741 #2(\@@_sort_clist:n{#3,#4})}
742 {\partial #3\c_math_superscript_token 2}\l_@@_Partial_end_tl
743 }
744 }
745 }
746 }

```

`\PartialSecondBigg` The `\PartialSecondBigg` macro and its starred variant replace `\left` and `\right` with `amsmath's` `\[module=amsmath]Biggl` and `\[module=amsmath]Biggr`.

```

747 \NewDocumentCommand{\PartialSecondBigg}{*}
748 { \tl_set:Nn \l_@@_Partial_start_tl {\biggl\l_@@_PartialOpen_tl}
749 \tl_set:Nn \l_@@_Partial_end_tl {\biggl\l_@@_PartialClose_tl}
750 \tl_set:Nn \l_@@_Partial_empty_end_tl {\biggl\l_@@_PartialClose_tl}
751 \PartialSecond
752 }

```

`\PartialSecondbigg` The `\PartialSecondbigg` macro and its starred variant replace `\left` and `\right` with `amsmath's` `\[module=amsmath]biggl` and `\[module=amsmath]biggr`.

```

753 \NewDocumentCommand{\PartialSecondbigg}{*}
754 { \tl_set:Nn \l_@@_Partial_start_tl {\biggl\l_@@_PartialOpen_tl}
755 \tl_set:Nn \l_@@_Partial_end_tl {\biggl\l_@@_PartialClose_tl}
756 \tl_set:Nn \l_@@_Partial_empty_end_tl {\biggl\l_@@_PartialClose_tl}
757 \PartialSecond
758 }

```

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

```

759 \bool_new:N \l_@@_has_x_or_y_bool
760 \NewDocumentCommand{\PartialMixSecond}{s m m m m}
761 {
762 \bool_if:nTF {#1}
763 {% Starred version
764 \settowidth{\l_@@_Partial_const_dim}{\ensuremath{#4}}%
765 \dim_add:Nn \l_@@_Partial_const_dim {-0.20\l_@@_Partial_const_dim}
766 \PartialMixSecond{#2}{#3}{#4}{#5}
767 \bool_if:nT \l_@@_subscripted_bool
768 { \dim_compare:nNnTF {\l_@@_operator_width_dim}
769 < {\l_@@_Partial_const_dim}
770 { \kern -\l_@@_operator_width_dim }
771 { \kern -\l_@@_Partial_const_dim }
772 }
773 }

```

```

774 {% Unstarred version
775   \bool_if:nTF \l_@@_subscripted_bool
776   {% subscripted version
777     \tl_if_eq:nnTF {#5} {}
778     {% Handle case of empty variables held constant
779       \l_@@_Partial_start_tl
780       \@@_frac:nn{\partial\c_math_superscript_token 2 #2}
781       {\partial #3\partial #4}\l_@@_Partial_empty_end_tl
782     }
783     { \l_@@_Partial_start_tl
784       \@@_frac:nn{\partial\c_math_superscript_token 2 #2}
785       {\partial #3\partial #4}\l_@@_Partial_end_tl
786       \c_math_subscript_token{#5}
787     }
788   }
789   {% not subscripted
790     \tl_if_eq:nnTF {#5} {}
791     {% empty argument
792       \l_@@_Partial_start_tl
793       \@@_frac:nn{\partial\c_math_superscript_token 2
794         #2(\@@_sort_clist:n{#3,#4,#5})}
795       {\partial #3\partial #4}\l_@@_Partial_empty_end_tl
796     }
797     {% Check whether #3 OR #4 are \Nt_i/etc.
798       \tl_if_in:nnTF {#3} {\Nt}
799       { \RenewDocumentCommand{\allbut}{0{j} m m}{\allcomponents{##3}}%
800         \l_@@_Partial_start_tl
801         \@@_frac:nn{\partial\c_math_superscript_token 2
802           #2(\@@_sort_clist:n{#4,#5})}
803         {\partial #3\partial #4}\l_@@_Partial_end_tl
804       }
805       { \tl_if_in:nnTF {#4} {\Nt}
806         { \RenewDocumentCommand{\allbut}{0{j} m m}{\allcomponents{##3}}%
807           \l_@@_Partial_start_tl
808           \@@_frac:nn{\partial\c_math_superscript_token 2
809             #2(\@@_sort_clist:n{#3,#5})}
810           {\partial #3\partial #4}\l_@@_Partial_end_tl
811         }
812         {% Check for x, y, or w
813           \bool_set_false:N \l_@@_has_x_or_y_bool
814           \tl_if_in:nnT {#3} {x}
815           { \l_@@_has_x_or_y_bool }
816           \tl_if_in:nnT {#3} {y}
817           { \l_@@_has_x_or_y_bool }
818           \tl_if_in:nnT {#3} {w}
819           { \l_@@_has_x_or_y_bool }
820           \bool_if:NTF \l_@@_has_x_or_y_bool
821           { \RenewDocumentCommand{\allbutlastand}{0{j} m m}
822             {\allcomponents{##3}}
823             \l_@@_Partial_start_tl
824             \@@_frac:nn{\partial\c_math_superscript_token 2
825               #2(\@@_sort_clist:n{#4,#5})}
826             {\partial #3\partial #4}\l_@@_Partial_end_tl
827           }
828           {
829             \l_@@_Partial_start_tl

```

```

830          \@@_frac:nn{\partial\c_math_superscript_token 2
831              #2(\@@_sort_clist:n{#3,#4,#5})}
832              {\partial #3\partial #4}\l_@@_Partial_end_tl
833          }
834      }
835  }
836 }
837 }
838 }
839 }

```

`\PartialMixSecondBigg` The macro `\PartialMixSecondBigg` is analogous to the aforementioned macros `\PartialMixSecondbigg`, `\PartialBigg` and `\PartialSecondBigg`. `\PartialMixSecondbigg` is analogous to `\Partialbigg` and `\PartialSecondbigg`.

```

840 \NewDocumentCommand{\PartialMixSecondBigg}{}
841 { \tl_set:Nn \l_@@_Partial_start_tl {\Biggl\l_@@_PartialOpen_tl}
842   \tl_set:Nn \l_@@_Partial_end_tl {\Biggl\l_@@_PartialClose_tl}
843   \tl_set:Nn \l_@@_Partial_empty_end_tl {\Biggl\l_@@_PartialClose_tl}
844   \PartialMixSecond
845 }
846 \NewDocumentCommand{\PartialMixSecondbigg}{}
847 { \tl_set:Nn \l_@@_Partial_start_tl {\biggl\l_@@_PartialOpen_tl}
848   \tl_set:Nn \l_@@_Partial_end_tl {\biggl\l_@@_PartialClose_tl}
849   \tl_set:Nn \l_@@_Partial_empty_end_tl {\biggl\l_@@_PartialClose_tl}
850   \PartialMixSecond
851 }

```

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

```

852 \AtBeginDocument{%
853   \ifpackageloaded{amsmath}{}{}%
854     \PackageWarningNoLine{thermodynamics}
855       {Package~amsmath~not~loaded;~load~to~make~PartialBigg~and~friends
856         ~work~correctly}%
857     \cs_set_eq:NN \PartialBigg \Partial
858     \cs_set_eq:NN \Partialbigg \Partial
859     \cs_set_eq:NN \PartialSecondBigg \PartialSecond
860     \cs_set_eq:NN \PartialSecondbigg \PartialSecond
861     \cs_set_eq:NN \PartialMixSecondBigg \PartialMixSecond
862     \cs_set_eq:NN \PartialMixSecondbigg \PartialMixSecond
863     \ProvideDocumentCommand{\rvert}{}{}{}
864     \ProvideDocumentCommand{\lvert}{}{}{}
865 }%
866 }

```

`\Partialinline` Inline partial derivatives.

```

\PartialSecondinline 867 \NewDocumentCommand{\Partialinline}{}
\PartialMixSecondinline 868 {
869   \cs_set:Nn \@@_frac:nn { ##1 \l_@@_Partial_middle_tl / ##2 }
870   \Partial
871 }
872 \NewDocumentCommand{\PartialSecondinline}{}
873 {
874   \cs_set:Nn \@@_frac:nn { ##1 \l_@@_Partial_middle_tl / ##2 }
875   \PartialSecond

```

```

876 }
877 \NewDocumentCommand{\PartialMixSecondinline}{}
878 {
879   \cs_set:Nn \l_@@_frac:nn { ##1 \l_@@_Partial_middle_tl / ##2 }
880   \PartialMixSecond
881 }

```

`\Partialinlinetext` Text-only (non-extensible delimiter) versions of `\Partialinline` and friends.

```

\PartialSecondinlinetext 882 \NewDocumentCommand{\Partialinlinetext}{}
\PartialMixSecondinlinetext 883 { \cs_set_eq:NN \l_@@_Partial_start_tl \l_@@_PartialOpen_tl
884   \cs_set_eq:NN \l_@@_Partial_end_tl \l_@@_PartialClose_tl
885   \cs_set_eq:NN \l_@@_Partial_empty_end_tl \l_@@_PartialEmptyClose_tl
886   \cs_set_eq:NN \l_@@_Partial_middle_tl \relax
887   \Partialinline
888 }
889 \NewDocumentCommand{\PartialSecondinlinetext}{}
890 { \cs_set_eq:NN \l_@@_Partial_start_tl \l_@@_PartialOpen_tl
891   \cs_set_eq:NN \l_@@_Partial_end_tl \l_@@_PartialClose_tl
892   \cs_set_eq:NN \l_@@_Partial_empty_end_tl \l_@@_PartialEmptyClose_tl
893   \cs_set_eq:NN \l_@@_Partial_middle_tl \relax
894   \PartialSecondinline
895 }
896 \NewDocumentCommand{\PartialMixSecondinlinetext}{}
897 { \cs_set_eq:NN \l_@@_Partial_start_tl \l_@@_PartialOpen_tl
898   \cs_set_eq:NN \l_@@_Partial_end_tl \l_@@_PartialClose_tl
899   \cs_set_eq:NN \l_@@_Partial_empty_end_tl \l_@@_PartialEmptyClose_tl
900   \cs_set_eq:NN \l_@@_Partial_middle_tl \relax
901   \PartialMixSecondinline
902 }

```

4.5 Local Override of Delimiters

We define several environments that *locally* override the delimiters on partial derivatives generated with `\Partial` and friends, the subscript notation for partial derivatives, and/or the definitions of range-oriented macros such as `\allNs`. These environments can be nested; the inner-most one will be honored if conflicts occur.

`thermoparentheses` (*env.*) Inside this environment, partial derivatives will have parentheses around them, regardless of package options.

```

903 \NewDocumentEnvironment{thermoparentheses}{}
904 { \cs_set:Nn \l_@@_PartialOpen_tl {(}
905   \cs_set:Nn \l_@@_PartialClose_tl {)}
906   \cs_set:Nn \l_@@_PartialEmptyClose_tl {)}
907 }{}

```

`thermobrackets` (*env.*) Inside this environment, partial derivatives will have brackets around them, regardless of package options.

```

908 \NewDocumentEnvironment{thermobrackets}{}
909 { \tl_set:Nn \l_@@_PartialOpen_tl {[}
910   \tl_set:Nn \l_@@_PartialClose_tl {]}
911   \tl_set:Nn \l_@@_PartialEmptyClose_tl {]}
912 }{}

```

`thermobraces` (*env.*) Inside this environment, partial derivatives will have braces around them, regardless of package options.


```

913 \NewDocumentEnvironment{thermobraces}{}
914 { \tl_set:Nn \l_@@_PartialOpen_tl {\}
915   \tl_set:Nn \l_@@_PartialClose_tl {\}
916   \tl_set:Nn \l_@@_PartialEmptyClose_tl {\}
917 }{}

```

`thermobar` (*env.*) Inside this environment, partial derivatives will have a trailing vertical bar, regardless of package options.

```

918 \NewDocumentEnvironment{thermobar}{}
919 { \tl_set:Nn \l_@@_PartialOpen_tl {.}
920   \tl_set:Nn \l_@@_PartialClose_tl {\rvert}
921   \tl_set:Nn \l_@@_PartialEmptyClose_tl {.}
922 }{}

```

`thermoplain` (*env.*) Inside this environment, partial derivatives will have no decorations around them, regardless of package options.

```

923 \NewDocumentEnvironment{thermoplain}{}
924 { \tl_set:Nn \l_@@_PartialOpen_tl {.}
925   \tl_set:Nn \l_@@_PartialClose_tl {.}
926   \tl_set:Nn \l_@@_PartialEmptyClose_tl {.}
927   \bool_set_false:N \l_@@_subscripted_bool
928 }{}

```

`thermoNOsubscripts` (*env.*) Inside this environment, subscripts will not be displayed to the right of partial derivatives, regardless of package options.

```

929 \NewDocumentEnvironment{thermoNOsubscripts}{}
930   {\bool_set_false:N \l_@@_subscripted_bool}
931   {}

```

`thermosubscripts` (*env.*) Inside this environment, subscripts will be displayed to the right of partial derivatives, regardless of package options.

```

932 \NewDocumentEnvironment{thermosubscripts}{}
933   {\bool_set_true:N \l_@@_subscripted_bool}
934   {}

```

`thermomolesrange` (*env.*) Inside this environment, the macro `\allNs` will expand to n_1, \dots, n_C (or equivalent symbols if `\Nt` and/or `\ncomponents` have been redefined), regardless of package options. Similar expansions will result for `\allXs`, `\allYs`, `\allMs`, and so on.

```

935 \NewDocumentEnvironment{thermomolesrange}{}
936   { \@@_set_moles_range }
937   {}

```

`thermoIntensiveplain` (*env.*)

```

938 % \begin{macrocode}
939 \NewDocumentEnvironment{thermoIntensiveplain}{}
940   { \@@_set_intensive_plain }
941   {}

```

`thermoExtensiveplain` (*env.*)

```

942 \NewDocumentEnvironment{thermoExtensiveplain}{}
943   { \@@_set_extensive_plain }
944   {}

```

thermointensivelowercase (*env.*)

```
945 \NewDocumentEnvironment{thermointensivelowercase}{}
946   {% {
947   \RenewExpandableDocumentCommand{\MacroFont}{}{
948   \fontencoding\encodingdefault
949   \fontfamily\ttdefault
950   \fontseries\mddefault
951   \fontshape\shapedefault
952   \footnotesize}
953
954   \@@_set_intensive_lowercase
955   \@@_set_lower_case_pms
956   }
957   {}
```

thermoextensivesuperscript (*env.*)

```
958 \NewDocumentEnvironment{thermoextensivesuperscript}{}
959   {
960   \@@_set_extensive_superscripts
961   }
962   {}
```

4.6 User-Interface Macros to Define Symbols

`\NewSubscriptedSymbol` First, we define a command that serves to create “subscripted” symbols; for example, typing `\cPi` 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.

```
963 \cs_new:Npn \@@_check_definable:nN #1#2
964 {
965   \bool_set_true:N \l_@@_arg_legal_bool
966   \tl_trim_spaces_apply:nN {#1} \tl_if_single_token:nTF
967   {
968     \str_set:Nx \l_tmp_str {\tl_to_str:n {#1}}
969     \int_compare:nNnT {\str_count:N \l_tmp_str} = 1
970     { \PackageError{thermodynamics}
971       {First~argument~of~'\tl_trim_spaces:o {\tl_to_str:n {#2}}}'~
972       must~be~a~command}
973     {The~first~argument~of~'\tl_trim_spaces:o {\tl_to_str:n {#2}}}'~
974     should~be~the~macro~that~will~be~used~to~refer~to~the~symbol.~
975     The~provided~argument~'\tl_trim_spaces:o {\tl_to_str:n {#1}}}'~
976     is~a~single~character.
977     \MessageBreak Perhaps~a~backslash~is~missing?}
978   }
979 }
980 { \PackageError{thermodynamics}
981   {First~argument~of~'\tl_trim_spaces:o {\tl_to_str:n {#2}}}'~
982   must~be~a~command}
983 {The~first~argument~of~'\tl_trim_spaces:o {\tl_to_str:n {#2}}}'~
984 should~be~the~macro~that~will~be~used~to~refer~to~the~symbol.~
985 The~provided~argument~'\tl_trim_spaces:o {\tl_to_str:n {#1}}}'~
986 contains~more~than~one~token.
987 \MessageBreak Perhaps~a~backslash~is~missing?}
988 }
989 }
990 \NewDocumentCommand{\NewSubscriptedSymbol}{m m m}
```

```

991 {
992   \@@_check_definable:nN {#1} \NewSubscriptedSymbol
993   \cs_if_exist:NT #1
994   { \PackageError{thermodynamics}
995     {Command~'\tl_trim_spaces:o {\tl_to_str:n {#1}}'~already~defined}
996     {You~have~used~
997       '\tl_trim_spaces:o {\tl_to_str:n {\NewSubscriptedSymbol}}'~
998       with~a~command~that~already~has~a~definition}
999   }
1000   \cs_new:cpn {\cs_to_str:N #1_one:n} ##1
1001   {
1002     {#2}\c_math_superscript_token{##1}
1003     \peek_catcode_remove:NTF \c_math_subscript_token
1004     { \use:c {\cs_to_str:N #1_three:n} }
1005     { \c_math_subscript_token{#3} }
1006   }
1007
1008   \cs_new:cpn {\cs_to_str:N #1_two:n} ##1
1009   { {#2}\c_math_subscript_token{#3,##1} }
1010
1011   \cs_new:cpn {\cs_to_str:N #1_three:n} ##1
1012   { \c_math_subscript_token{#3,##1} }
1013
1014   \NewDocumentCommand{#1}{}
1015   {% @branch
1016     \peek_catcode_remove:NTF \c_math_superscript_token
1017     { \use:c {\cs_to_str:N #1_one:n} }
1018     { \peek_catcode_remove:NTF \c_math_subscript_token
1019       { \use:c {\cs_to_str:N #1_two:n} }
1020       { {#2}\c_math_subscript_token{#3} }
1021     }
1022   }
1023 }
1024 \NewDocumentCommand{\RenewSubscriptedSymbol}{m m m}
1025 {
1026   \@@_check_definable:nN {#1} \RenewSubscriptedSymbol
1027   \cs_if_exist:NF #1
1028   { \PackageError{thermodynamics}
1029     {Command~'\tl_trim_spaces:o {\tl_to_str:n {#1}}'~not~defined}
1030     {You~have~used~
1031       '\tl_trim_spaces:o {\tl_to_str:n {\RenewSubscriptedSymbol}}'~
1032       with~a~command~that~does~not~have~a~definition}
1033   }
1034   \cs_set:cpn {\cs_to_str:N #1_one:n} ##1
1035   {
1036     {#2}\c_math_superscript_token{##1}
1037     \peek_catcode_remove:NTF \c_math_subscript_token
1038     { \use:c {\cs_to_str:N #1_three:n} }
1039     { \c_math_subscript_token{#3} }
1040   }
1041
1042   \cs_set:cpn {\cs_to_str:N #1_two:n} ##1
1043   { {#2}\c_math_subscript_token{#3,##1} }
1044
1045   \cs_set:cpn {\cs_to_str:N #1_three:n} ##1
1046   { \c_math_subscript_token{#3,##1} }

```

```

1047
1048 \RenewDocumentCommand{#1}{}
1049 {% @branch
1050   \peek_catcode_remove:NTF \c_math_superscript_token
1051   { \use:c {\cs_to_str:N #1_one:n} }
1052   { \peek_catcode_remove:NTF \c_math_subscript_token
1053     { \use:c {\cs_to_str:N #1_two:n} }
1054     { {#2}\c_math_subscript_token{#3} } }
1055   }
1056 }
1057
1058 }

```

\NewSuperscriptedSymbol

```

1059 \NewDocumentCommand{\NewSuperscriptedSymbol}{m m m}
1060 {
1061   \@@_check_definable:nN {#1} \NewSuperscriptedSymbol
1062   \cs_if_exist:NT #1
1063   { \PackageError{thermodynamics}
1064     {Command~'\tl_trim_spaces:o {\tl_to_str:n {#1}}'~already~defined}
1065     {You~have~used~
1066       '\tl_trim_spaces:o {\tl_to_str:n {\NewSuperscriptedSymbol}}'~
1067       with~a~command~that~already~has~a~definition}
1068   }
1069   \cs_new:cpn {\cs_to_str:N #1_one:n} ##1
1070   {
1071     {#2}\c_math_subscript_token{##1}
1072     \peek_catcode_remove:NTF \c_math_superscript_token
1073     { \use:c {\cs_to_str:N #1_three:n} }
1074     { \c_math_superscript_token{#3} }
1075   }
1076
1077   \cs_new:cpn {\cs_to_str:N #1_two:n} ##1
1078   { {#2}\c_math_superscript_token{#3,##1} }
1079
1080   \cs_new:cpn {\cs_to_str:N #1_three:n} ##1
1081   { {#2}\c_math_superscript_token{#3,##1} }
1082
1083   \NewDocumentCommand{#1}{}
1084   {
1085     \peek_catcode_remove:NTF \c_math_subscript_token
1086     { \use:c {\cs_to_str:N #1_one:n} }
1087     { \peek_catcode_remove:NTF \c_math_superscript_token
1088       { \use:c {\cs_to_str:N #1_two:n} }
1089       { {#2}\c_math_superscript_token{#3} } }
1090   }
1091 }
1092 }
1093 \NewDocumentCommand{\RenewSuperscriptedSymbol}{m m m}
1094 {
1095   \@@_check_definable:nN {#1} \RenewSuperscriptedSymbol
1096   \cs_if_exist:NF #1
1097   { \PackageError{thermodynamics}
1098     {Command~'\tl_trim_spaces:o {\tl_to_str:n {#1}}'~not~defined}
1099     {You~have~used~
1100       '\tl_trim_spaces:o {\tl_to_str:n {\RenewSuperscriptedSymbol}}'~

```

```

1101         with~a~command~that~does~not~have~a~definition}
1102     }
1103     \cs_set:cpn {\cs_to_str:N #1_one:n} ##1
1104     {
1105         {#2}\c_math_subscript_token{##1}
1106         \peek_catcode_remove:NTF \c_math_superscript_token
1107         { \use:c {\cs_to_str:N #1_three:n} }
1108         { \c_math_superscript_token{#3} }
1109     }
1110
1111     \cs_set:cpn {\cs_to_str:N #1_two:n} ##1
1112     { {#2}\c_math_superscript_token{#3,##1} }
1113
1114     \cs_set:cpn {\cs_to_str:N #1_three:n} ##1
1115     { \c_math_superscript_token{#3,##1} }
1116
1117     \RenewDocumentCommand{#1}{}
1118     {% @branch
1119         \peek_catcode_remove:NTF \c_math_subscript_token
1120         { \use:c {\cs_to_str:N #1_one:n} }
1121         { \peek_catcode_remove:NTF \c_math_superscript_token
1122           { \use:c {\cs_to_str:N #1_two:n} }
1123           { {#2}\c_math_superscript_token{#3} }
1124         }
1125     }
1126 }

```

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

```

\compressibilitysymbol 1127 \NewExpandableDocumentCommand{\heatcapacitiesymbol}{}{C}
\expansivitysymbol    1128 \NewExpandableDocumentCommand{\compressibilitysymbol}{}{\kappa}
                       1129 \NewExpandableDocumentCommand{\expansivitysymbol}{}{\alpha}
                       1130 \NewExpandableDocumentCommand{\JTsymbolsymbol}{}{\mu}

```

`\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”: `\cVi` 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.

```

1131 \NewSubscriptedSymbol{\cV}{\@@_intensive:n \heatcapacitiesymbol}
1132     {\g_@@_volume_symbol}
1133 \NewSubscriptedSymbol{\cP}{\@@_intensive:n \heatcapacitiesymbol}
1134     {\g_@@_pressure_symbol}

```

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

```

\cPt 1135 \NewSubscriptedSymbol{\cVt}{\@@_extensive:n \heatcapacitiesymbol}
      1136     {\g_@@_volume_symbol}
      1137 \NewSubscriptedSymbol{\cPt}{\@@_extensive:n \heatcapacitiesymbol}
      1138     {\g_@@_pressure_symbol}

```

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

```

\cPs 1139 \NewSubscriptedSymbol{\cVs}{\@@_specific:n \heatcapacitiesymbol}
      1140     {\g_@@_volume_symbol}
      1141 \NewSubscriptedSymbol{\cPs}{\@@_specific:n \heatcapacitiesymbol}
      1142     {\g_@@_pressure_symbol}

```

`\kappaT` The isothermal and adiabatic compressibilities are defined similarly, but those do not
`\kappaS` have extensive versions for obvious reasons.

```
1143 \NewSubscriptedSymbol{\kappaT}{\compressibilitysymbol}
1144     {\g_@@_temperature_symbol}
1145 \NewSubscriptedSymbol{\kappaS}{\compressibilitysymbol}
1146     {\g_@@_entropy_symbol}
```

`\alphaP` The macro `\alphaP` is intended to refer to the isobaric volume expansivity, while
`\alphaS` `\alphaS` is the isentropic volume expansivity.

```
1147 \NewSubscriptedSymbol{\alphaP}{\expansivitysymbol}{\g_@@_pressure_symbol}
1148 \NewSubscriptedSymbol{\alphaS}{\expansivitysymbol}{\g_@@_entropy_symbol}
```

`\muJT` The macro `\muJT` renders the Joule–Thomson coefficient.

```
1149 \NewSubscriptedSymbol{\muJT}{\JTsymbol}{JT}
```

`\Psat` The `\Psat` macro (and its clone, the `\Pvap` macro) should be used for the saturation
`\Pvap` pressure. Similarly, a `\phisat` macro typesets the fugacity coefficient at saturation.

`\phisat` The `\fsat` macro similarly renders the fugacity at saturation. Other saturation prop-
`\fsat` erties should use M^{sat} or similar, preferably by defining another macro such as
`\sat` `\Gmsat`.

```
1150 \NewDocumentCommand{\sat}{}{\text{sat}}
1151 \NewDocumentCommand{\Psat}{}{P\c_math_superscript_token\sat}
1152 \NewDocumentCommand{\Pvap}{}{\Psat}
1153 \NewDocumentCommand{\phisat}{}{\phi\c_math_superscript_token\sat}
1154 \NewDocumentCommand{\fsat}{}{\fpure\c_math_superscript_token\sat}
```

`\std` The `\std` macro denotes standard properties. `\Pstd` and `\fstd` are defined for con-
`\Pstd` venience and for consistency across textbooks.

```
\fstd 1155 \NewDocumentCommand{\std}{}{\circ}
1156 \NewDocumentCommand{\Pstd}{}{P\c_math_superscript_token \std}
1157 \NewDocumentCommand{\fstd}{}{f\c_math_superscript_token \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.

```
1158 \NewDocumentCommand{\mixing}{}{\text{mix}}
1159 \NewDocumentCommand{\Deltamix}{m}
1160     {\Delta\c_math_subscript_token\mixing}
```

`\Deltafus` Similar entities for property changes on fusion, reaction, sublimation, and vaporiza-
`\fusion` tion are defined.

```
\Deltavap 1161 \NewDocumentCommand{\fusion}{}{\text{fus}}
\vaporization 1162 \NewDocumentCommand{\reaction}{}{\text{rxn}}
\Deltafus 1163 \NewDocumentCommand{\sublimation}{}{\text{sub}}
\sublimation 1164 \NewDocumentCommand{\vaporization}{}{\text{vap}}
\Deltafus 1165 \NewDocumentCommand{\formation}{}{f}
\Deltafus 1166 \NewDocumentCommand{\Deltafus}{m}
\reaction 1167     {\Delta #1\c_math_superscript_token\fusion}
1168 \NewDocumentCommand{\Deltasub}{m}
1169     {\Delta #1\c_math_superscript_token\sublimation}
1170 \NewDocumentCommand{\Deltavap}{m}
1171     {\Delta #1\c_math_superscript_token\vaporization}
1172 \NewDocumentCommand{\Deltarxn}{m}
```

```

1173     {\Delta #1\c_math_subscript_token\reaction}
1174 \tl_new:N \l_@@_Deltaf_sym_tl
1175 \NewSubscriptedSymbol{\@@_Deltaf}{\l_@@_Deltaf_sym_tl}{f}
1176 \NewDocumentCommand{\Deltaf}{m}
1177 { \tl_set:Nn \l_@@_Deltaf_sym_tl {\Delta #1}
1178   \@@_Deltaf
1179 }

```

`\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.

```

1180 \NewDocumentCommand{\fmix}{}{\hat{f}}

```

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

```

1181 \NewDocumentCommand{\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, `\hipure`, for fugacity coefficients.

```

1182 \NewDocumentCommand{\fpure}{}{f}
1183 \NewDocumentCommand{\hipure}{}{\phi}

```

4.7 Partial Molar Quantities

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

```

1184 \tl_new:N \l_@@_pm_symbol_tl
1185 \tl_new:N \l_@@_pm_arg_tl
1186 \NewDocumentCommand{\partialmolar}{m}
1187 {
1188   \tl_set:Nn \l_@@_pm_symbol_tl {#1}
1189   \@@_generic_pm:
1190 }
1191 %% cases to consider:
1192 %% (1) \Mpm{i}
1193 %% (2) \Mpm[S]{i}
1194 %% (3) \Mpm^S_i
1195 %% (4) \Mpm_i^S
1196 %% (5) \Mpm_i
1197 %% note that \Mpm^S with no subscript makes no sense and is thus forbidden
1198 \cs_new:Nn \@@_generic_pm:
1199 {
1200   \peek_catcode_remove:NTF \c_math_subscript_token
1201   {% case 4 or case 5

```

```

1202   \@@_pm_case_four_or_five
1203   }
1204   {% look for superscript token
1205   \peek_catcode_remove:NTF \c_math_superscript_token
1206   {% case 3: \Mpm^{#1}_{#2} or \Mpm^{#1}{#2}
1207   \@@_pm_case_three
1208   }
1209   {% Look for optional argument [...]
1210   \peek_charcode:NTF [
1211   {% case 2: \Mpm[S]{i}
1212   \@@_pm_case_two
1213   }
1214   {% case 1: \Mpm{i}
1215   \@@_pm_case_one
1216   }
1217   }
1218   }
1219 }
1220 \cs_new:Npn \@@_pm_case_one #1
1221 {
1222   \@@_overline:n {\l_@@_pm_symbol_tl\c_math_subscript_token{#1}}
1223 }
1224 \cs_new:Npn \@@_pm_case_two [#1]#2
1225 {
1226   \@@_overline:n {\l_@@_pm_symbol_tl
1227     \c_math_superscript_token{#1}\c_math_subscript_token{#2}}
1228 }
1229 \cs_new:Npn \@@_pm_case_three #1
1230 {
1231   \tl_set:Nn \l_@@_pm_arg_tl {#1}
1232   \peek_catcode_remove:NTF \c_math_subscript_token
1233   { \@@_pm_case_three_part_two }
1234   { \@@_pm_case_three_part_two }
1235 }
1236 \cs_new:Npn \@@_pm_case_three_part_two #1
1237 {
1238   \@@_overline:n {\l_@@_pm_symbol_tl
1239     \c_math_superscript_token{\l_@@_pm_arg_tl}
1240     \c_math_subscript_token{#1}}
1241 }
1242 \cs_new:Npn \@@_pm_case_four_or_five #1
1243 {
1244   \tl_set:Nn \l_@@_pm_arg_tl {#1}
1245   \peek_catcode_remove:NTF \c_math_superscript_token
1246   { \@@_pm_case_four }
1247   { \@@_pm_case_five }
1248 }
1249 \cs_new:Npn \@@_pm_case_four #1
1250 {
1251   \@@_overline:n {\l_@@_pm_symbol_tl\c_math_superscript_token{#1}
1252     \c_math_subscript_token{\l_@@_pm_arg_tl}}
1253 }
1254 \cs_new:Npn \@@_pm_case_five
1255 {
1256   \@@_overline:n {\l_@@_pm_symbol_tl
1257     \c_math_subscript_token{\l_@@_pm_arg_tl}}

```


4.8 Symbol Definitions

These macros define the user interface to the symbols for energy, volume, and so forth. There are five commands that define thermodynamic properties.

`\NewExtensiveProperty` The command `\NewExtensiveProperty` declares macros for a total, molar, and specific version of the symbol; for example, a second heat-like property could be defined via

```
\NewExtensiveProperty{R}{\mathcal{Q}}
```

The command above would declare the macros `\Rt`, `\Rm`, and `\Rs` that expand to \underline{Q} , Q , and \hat{Q} , respectively, using the default package options.

`\NewPartialMolarProperty` The command `\NewPartialMolarProperty` declares a macro for the partial molar quantity. For example,

```
\NewPartialMolarProperty{I}{\Psi}
```

would create the command `\Ipm`, which would typeset a partial molar command with the base symbol Ψ , yielding $\bar{\Psi}_i$.

`\NewThermodynamicProperty` Declaring a new potential is handled by the `\NewThermodynamicProperty` macro, which takes two arguments. The first is the base of the name, and the second is the base of the symbol. This declares four new commands for the extensive, molar, specific, and partial molar properties. These commands consist of the first argument followed by `t`, `m`, `s`, and `pm`, respectively. For example, one might define the entropy via

```
\NewThermodynamicProperty{S}{S}
```

and it would define the macros `\St`, `\Sm`, `\Ss`, and `\Spm` that yield, respectively, \underline{S} , S , \hat{S} , and \bar{S}_i (assuming the subscript to the partial molar quantity was i). It would also declare residual and excess properties for that base symbol. Note that the actual definition of the entropy and the other standard properties is slightly more complicated so as to allow for different symbols to be used in different textbooks.

```
1259 \NewDocumentCommand{\NewThermodynamicProperty}{m m}
1260 {
1261   \NewExtensiveProperty{#1}{#2}
1262   \NewPartialMolarProperty{#1}{#2}
1263   \NewResidualProperty{#1}{#2}
1264   \NewExcessProperty{#1}{#2}
1265 }
1266 \NewDocumentCommand{\NewExtensiveProperty}{m m}
1267 {
1268   % Extensive property
1269   \exp_after:wN \NewDocumentCommand \exp_after:wN
1270     {\cs:w #1t\cs_end:}{\@@_extensive:n {#2}}
1271   % Molar property
1272   \exp_after:wN \NewDocumentCommand \exp_after:wN
1273     {\cs:w #1m\cs_end:}{\@@_intensive:n {#2}}
1274   % Specific property
1275   \exp_after:wN \NewDocumentCommand \exp_after:wN
1276     {\cs:w #1s\cs_end:}{\@@_specific:n {#2}}
1277 }
```

```

1278 \NewDocumentCommand{\NewPartialMolarProperty}{m m}
1279 {
1280 % Partial molar property
1281 \exp_after:wN \NewDocumentCommand \exp_after:wN
1282   {\cs:w #1pm\cs_end:}{}{\partialmolar{#2}}
1283 }
1284 \NewDocumentCommand{\NewExcessProperty}{m m}
1285 {
1286 \exp_after:wN \NewSuperscriptedSymbol \exp_after:wN
1287   {\cs:w #1 Et\cs_end:}{\@@_extensive:n{#2}}{\excess}
1288 \exp_after:wN \NewSuperscriptedSymbol \exp_after:wN
1289   {\cs:w #1 E\cs_end:}{\@@_intensive:n{#2}}{\excess}
1290 \exp_after:wN \NewSuperscriptedSymbol \exp_after:wN
1291   {\cs:w #1 Es\cs_end:}{\@@_specific:n{#2}}{\excess}
1292
1293 %^^A TODO: make super- or subscripted partial molar quantities work
1294 % Excess partial molar property
1295 \exp_after:wN \NewDocumentCommand \exp_after:wN
1296   {\cs:w #1Epm\cs_end:}{}{\partialmolar{#2}
1297     \c_math_superscript_token\excess}
1298 }
1299 \NewDocumentCommand{\NewResidualProperty}{m m}
1300 {
1301 \exp_after:wN \NewSuperscriptedSymbol \exp_after:wN
1302   {\cs:w #1 Rt\cs_end:}{\@@_extensive:n{#2}}{\residual}
1303 \exp_after:wN \NewSuperscriptedSymbol \exp_after:wN
1304   {\cs:w #1 R\cs_end:}{\@@_intensive:n{#2}}{\residual}
1305 \exp_after:wN \NewSuperscriptedSymbol \exp_after:wN
1306   {\cs:w #1 Rs\cs_end:}{\@@_specific:n{#2}}{\residual}
1307
1308 % Residual partial molar property
1309 \exp_after:wN \NewDocumentCommand \exp_after:wN
1310   {\cs:w #1Rpm\cs_end:}{}{\partialmolar{#2}
1311     \c_math_superscript_token\residual}
1312 }

```

\Nt We define \Nt as the number of moles, as that changes between books a lot, but it \Et does not have extensive, molar, and specific equivalents. Heat and work are defined, \Em but lack partial molar properties. Area does not have excess or residual properties.

```

\Es 1313 \NewDocumentCommand{\Nt}{}{\g_@@_mole_symbol}
\Epm 1314 \NewThermodynamicProperty{E}{\g_@@_total_energy_symbol}
1315 \NewThermodynamicProperty{U}{\g_@@_internal_energy_symbol}
1316 \NewThermodynamicProperty{F}{\g_@@_Helmholtz_symbol}
1317 \NewThermodynamicProperty{G}{\g_@@_Gibbs_symbol}
1318 \NewThermodynamicProperty{H}{\g_@@_enthalpy_symbol}
1319 \NewThermodynamicProperty{L}{\g_@@_Landau_symbol}
1320 \NewThermodynamicProperty{V}{\g_@@_volume_symbol}
1321 \NewThermodynamicProperty{S}{\g_@@_entropy_symbol}
1322 \NewExtensiveProperty{A}{\g_@@_area_symbol}
1323 \NewPartialMolarProperty{A}{\g_@@_area_symbol}
1324 \NewExtensiveProperty{Q}{\g_@@_heat_symbol}
1325 \NewExtensiveProperty{W}{\g_@@_work_symbol}

```

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

```

1326 \NewDocumentCommand{\cPpm}{}

```

```

1327 {
1328 \cs_set:Npn \@@_pm_case_one ##1
1329 {
1330 \@@_overline:n {\l_@@_pm_symbol_tl\c_math_subscript_token
1331 {g_@@_pressure_symbol,##1}}
1332 }
1333 \cs_set:Npn \@@_pm_case_two [##1]##2
1334 {
1335 \@@_overline:n {\l_@@_pm_symbol_tl\c_math_superscript_token{##1}
1336 \c_math_subscript_token{g_@@_pressure_symbol,##2}}
1337 }
1338 \cs_set:Npn \@@_pm_case_three_part_two ##1
1339 {
1340 \@@_overline:n {\l_@@_pm_symbol_tl
1341 \c_math_superscript_token{\l_@@_pm_arg_tl}
1342 \c_math_subscript_token{g_@@_pressure_symbol,##1}}
1343 }
1344 \cs_set:Npn \@@_pm_case_four ##1
1345 {
1346 \@@_overline:n {\l_@@_pm_symbol_tl
1347 \c_math_superscript_token{##1}\c_math_subscript_token
1348 {g_@@_pressure_symbol,\l_@@_pm_arg_tl}}
1349 }
1350 \cs_set:Npn \@@_pm_case_five
1351 {
1352 \@@_overline:n {\l_@@_pm_symbol_tl\c_math_subscript_token
1353 {g_@@_pressure_symbol,\l_@@_pm_arg_tl}}
1354 }
1355 \partialmolar{heatcapacitiesymbol}
1356 }
1357 \NewDocumentCommand{\cVpm}{}
1358 {
1359 \cs_set:Npn \@@_pm_case_one ##1
1360 {
1361 \@@_overline:n {\l_@@_pm_symbol_tl\c_math_subscript_token
1362 {g_@@_volume_symbol,##1}}
1363 }
1364 \cs_set:Npn \@@_pm_case_two [##1]##2
1365 {
1366 \@@_overline:n {\l_@@_pm_symbol_tl
1367 \c_math_superscript_token{##1}\c_math_subscript_token
1368 {g_@@_volume_symbol,##2}}
1369 }
1370 \cs_set:Npn \@@_pm_case_three_part_two ##1
1371 {
1372 \@@_overline:n {\l_@@_pm_symbol_tl
1373 \c_math_superscript_token{\l_@@_pm_arg_tl}
1374 \c_math_subscript_token{g_@@_volume_symbol,##1}}
1375 }
1376 \cs_set:Npn \@@_pm_case_four ##1
1377 {
1378 \@@_overline:n {\l_@@_pm_symbol_tl
1379 \c_math_superscript_token{##1}
1380 \c_math_subscript_token
1381 {g_@@_volume_symbol,\l_@@_pm_arg_tl}}
1382 }

```

```

1383 \cs_set:Npn \@@_pm_case_five
1384 {
1385   \@@_overline:n {\l_@@_pm_symbol_tl\c_math_subscript_token
1386     {g_@@_volume_symbol,\l_@@_pm_arg_tl}}
1387 }
1388 \partialmolar{\heatcapacitiesymbol}
1389 }

```

4.9 Residual and Excess Properties

`\residual` Macros are defined for residual properties (departure from 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.

```

1390 \NewDocumentCommand{\residual}{}{R}
1391 \NewDocumentCommand{\excess}{}{E}

```

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

```

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

```

gives

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

```

1392 \NewDocumentCommand{\sumall}{m m}
1393 { \sum\c_math_subscript_token{#2=1}
1394   \c_math_superscript_token{\ncomponents} }
1395 \NewDocumentCommand{\sumallbutlast}{m m}
1396 { \sum\c_math_subscript_token{#2=1}
1397   \c_math_superscript_token{\ncomponents-1} }
1398 \NewDocumentCommand{\prodall}{m m}
1399 { \prod\c_math_subscript_token{#2=1}
1400   \c_math_superscript_token{\ncomponents} }

```

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

```

1401 \NewDocumentCommand{\IG}{}{\text{IG}}
1402 \NewDocumentCommand{\IGM}{}{\text{IGM}}
1403 \NewDocumentCommand{\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.

```

1404 \NewDocumentCommand{\Henryrat}{}{h}
1405 \NewDocumentCommand{\Henrymol}{}{\mathcal{H}}

```

`\gammarat` The ordinary activity coefficient is universally denoted γ , so I have not defined a `\gammamol` special macro for that. However, symbols for the Henry’s Law activity coefficients are far from universal, so I have defined macros to make their use consistent. The

defaults render `\gammamat` 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.

```

1406 \AtBeginDocument{%
1407   \providecommand*\square{%
1408     \text{\leavevmode
1409       \hbox to.65em{%
1410         \hfil\vrule
1411         \vbox to.53em{\hrule width.45em\vfil\hrule}%
1412         \vrule\hfil}%
1413     }%
1414 }%
1415 }
1416 \NewDocumentCommand{\gammamat}{}{\gamma\c_math_superscript_token\ast}
1417 \NewDocumentCommand{\gammamol}{}{\gamma\c_math_superscript_token\square}

```

`thermovmatrix` (*env.*) This is a non-user-interface wrapper environment used to detect, in effect, whether `amsmath` has been loaded. If so, it uses its `vmatrix` environment for Jacobians; if not, it fakes it with `array` (which does not look nearly as good).

```

1418 \NewDocumentEnvironment{thermovmatrix}{}
1419 { \cs_if_exist:NTF \vmatrix
1420   { \begin{vmatrix} }
1421   { \left|\begin{array}{c c c c c c c c c c} }
1422 }
1423 { \cs_if_exist:NTF \endvmatrix
1424   { \end{vmatrix} }
1425   { \end{array}\right| }
1426 }

```

`Jacobian` The `\Jacobian` command typesets the Leibnitz notation for the Jacobian determinant.

```

1427 \NewDocumentCommand{\Jacobian}{m m}
1428   {\@@_frac:nn{\partial(#1)}{\partial{(#2)}}}

```

`Jacobiandet` Similarly, the `\Jacobiandet` macro typesets the actual determinant that the Jacobian notation represents.

```

1429 \NewDocumentCommand{\Jacobiandet}{0{} 0{} m m}
1430 {
1431   \@@_Jacobian_set_ncomponents:nn {#3} {#4}
1432   \begin{thermovmatrix}
1433     \@@_Jacobianmatrix:nnnn {#1} {#2} {#3} {#4}
1434   \end{thermovmatrix}
1435 }
1436 \seq_new:N \l_@@_row_seq
1437 \seq_new:N \l_@@_matrix_seq
1438 \clist_new:N \l_@@_other_vars_clist
1439 \clist_new:N \l_@@_other_vars_copy_clist
1440 \tl_new:N \l_@@_Jacobian_x_tl
1441 \tl_new:N \l_@@_Jacobian_n_tl
1442 \tl_new:N \l_@@_Jacobian_temp_tl
1443 \bool_new:N \l_@@_found_dots_bool
1444 \cs_new:Nn \@@_Jacobian_set_ncomponents:nn
1445 {
1446   % If any entry is \dots, we assume the Jacobian is of the form
1447   % d(f_1,\dots,f_n)/d(x_1,\dots,x_n) where f is some function
1448   % (any symbol) and x is some variable (any symbol).

```

```

1449 \tl_if_in:nnTF {#1} {\dots}
1450 {% Has dots
1451   \bool_set_true:N \l_@@_found_dots_bool
1452   % look for what "x" is
1453   \tl_set:Nn \l_@@_Jacobian_x_tl {\tl_head:n {#2}}
1454   % look for what "n" is and set \ncomponents to it
1455   \tl_set:Nx \l_@@_Jacobian_n_tl {\tl_item:nn {#2} {-1}}
1456   \RenewExpandableDocumentCommand{\ncomponents}{}{\l_@@_Jacobian_n_tl}
1457 }
1458 {% Does not have dots; proceed accordingly
1459   \bool_set_false:N \l_@@_found_dots_bool
1460 }
1461 }
1462 \cs_new_protected:Nn \@@_Jacobianmatrix:nnnn
1463 {
1464   \seq_clear:N \l_@@_matrix_seq
1465   \clist_set:Nn \l_@@_other_vars_clist {#4}
1466   \clist_set_eq:NN \l_@@_other_vars_copy_clist \l_@@_other_vars_clist
1467
1468   \clist_map_inline:nn {#3}
1469   {
1470     \seq_clear:N \l_@@_row_seq
1471     \tl_if_in:nnTF {##1} {\dots}
1472     {% The current row has "dots" => row is \vdots && \vdots
1473       \seq_put_right:Nn \l_@@_matrix_seq
1474         { \vdots \c_alignment_token \c_alignment_token \vdots }
1475     }
1476     {% Ordinary row
1477       \clist_map_inline:nn {#4}
1478       {
1479         \tl_if_in:nnTF {####1} {\dots}
1480         {% this column has "dots" in it
1481           \seq_put_right:Nn \l_@@_row_seq \dots
1482         }
1483         {% Normal column
1484           \clist_set_eq:NN \l_@@_other_vars_clist
1485             \l_@@_other_vars_copy_clist
1486           \clist_remove_all:Nn \l_@@_other_vars_clist {####1}
1487           \bool_if:NTF \l_@@_found_dots_bool
1488           { \tl_set:Nn \l_@@_Jacobian_temp_tl {\tl_item:nn {####1} {-1}}
1489             \seq_put_right:Nx \l_@@_row_seq
1490             {
1491               #1\Partial{##1}{####1}
1492               {\allbut{\l_@@_Jacobian_temp_tl}{\l_@@_Jacobian_x_tl}}
1493             }
1494           }
1495           { \seq_put_right:Nx \l_@@_row_seq
1496             {
1497               #1\Partial{##1}{####1}
1498               {\clist_use:Nn \l_@@_other_vars_clist ,}
1499             }
1500           }
1501         }
1502       }
1503     \seq_put_right:Nx \l_@@_matrix_seq
1504     {

```

```

1505     \seq_use:Nn \l_@@_row_seq { \c_alignment_token }
1506   }
1507 }
1508 }
1509 \tl_if_empty:nTF {#2}
1510 {
1511   \tl_if_eq:nnTF {#1} {\displaystyle}
1512   { \seq_use:Nn \l_@@_matrix_seq { \[2.75ex] } }
1513   { \seq_use:Nn \l_@@_matrix_seq { \[1.25ex] } }
1514 }
1515 {
1516   \seq_use:Nn \l_@@_matrix_seq { \[#2] }
1517 }
1518 }

```

Change History

v1.00	to handle an arbitrary number of variables and implemented	
General: Initial public release	\Jacobiandet to handle the matrix representation of the Jacobian. . . .	1
v1.01		
General: Added \Partialinline and friends to facilitate in-line (non-display-mode) partial derivatives, with corresponding changes to \Partial and friends for ease of implementation. Also added \Partialinlinetext and friends for non-expanding delimiters.	\NewSubscriptedSymbol: Deleted \DeclareSubscriptedSymbol in favor of xparse-based \NewSubscriptedSymbol and \RenewSubscriptedSymbol.	39
Changed options with two <i>E</i> or two <i>A</i> variables to use calligraphic letters for the less-common of the two.	\NewSuperscriptedSymbol: Created \NewSuperscriptedSymbol to handle superscripted excess and residual properties without intervention.	42
\allcomponents: Updated \allcomponents to include an optional argument that changes N_i to N_j , say, when using TesterModell or other options that denote moles of all components that way. Similar updates to \allNs and friends.	\Partial: Changed length added to \l_@@_Partial_const_dim from -0.15 to -0.20	23
\Partial: Changed \adjust@width 0.1\operator@width (from 2pt)	thermoextensiveplain: Added environment to invoke the “extensive-plain” option locally. . . .	35
Changed \adjust@width to 2pt (up from 1pt)	thermoextensivesuperscript: Added environment to invoke the “extensive-superscript” option locally.	35
v2.00	thermointensivelowercase: Added environment to invoke the “intensive-lowercase” option locally.	41
General: Added \muJT to represent Joule–Thomson coefficients (which have different notation across textbooks).	thermointensivelowercase: Added environment to invoke the “intensive-lowercase” option locally.	6
Revision to use L ^A T _E X ₃ (expl3/xparse) syntax layer	thermomolesrange: Added environment to invoke the “moles-range” definitions of \allNs and friends locally.	1
Jacobian: Reimplemented \Jacobian		41

Index

Numbers written in italic 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.

A		\EEpm	4
\allbut	13, <u>257</u>	\EEs	4
\allbutlastand	13, <u>257</u>	\EEt	4
\allcomponents	<u>257</u>	\Em	4, <u>1313</u>
\allMs	12, <u>245</u>	environments:	
\allMsbut	12, <u>251</u>	thermobar	18, <u>918</u>
\allmus	12, <u>245</u>	thermobraces	<u>913</u>
\allmusbut	12	thermobrackets	18, <u>908</u>
\allNs	12, 18, 19, <u>245</u>	thermoextensiveplain	17, <u>942</u>
\allNsbut	12, 18, 19, <u>251</u>	thermoextensivesuperscript	17, <u>958</u>
\allWs	12, <u>245</u>	thermointensivelowercase	17, <u>945</u>
\allWsbut	12, <u>251</u>	thermointensivplain	17, <u>938</u>
\allXs	12, 19, <u>245</u>	thermomolesrange	18, <u>935</u>
\allXsbut	12, 19, <u>251</u>	thermoNOsubscripts	18, <u>929</u>
\allYs	12, <u>245</u>	thermoparentheses	18, <u>903</u>
\allYsbut	12, <u>251</u>	thermoplain	18, <u>923</u>
\alphaP	5, 6, <u>1147</u>	thermosubscripts	18, <u>932</u>
\alphaS	5, 6, <u>1147</u>	thermovmatrix	<u>1418</u>
\Am	4	\Epm	4, <u>1313</u>
\amix	5	\ER	4
\Apm	4	\ERpm	4
\As	4	\ERs	4
\At	4	\ERT	4
C		\Es	4, <u>1313</u>
\compressibilitysymbol	5, <u>1127</u>	\Et	4, <u>1313</u>
\cP	3, 5, 6, <u>1131</u>	\excess	9, <u>1390</u>
\cPpm	3, <u>1326</u>	\expansivitysymbol	5, <u>1127</u>
\cPs	3, <u>1139</u>	F	
\cPt	3, <u>1135</u>	\FE	4
\cV	3, 5, 6, <u>1131</u>	\FEpm	4
\cVpm	3, <u>1326</u>	\FEs	4
\cVs	3, <u>1139</u>	\FEt	4
\cVt	3, <u>1135</u>	\Fm	4
D		\fmix	5, 6, <u>1180</u>
\dbar	4, 18	\fpm	4
\Deltaf	8	\fpure	5, 6, <u>1182</u>
\Deltafus	5, 8, <u>1161</u>	\FR	4
\Deltamix	5, 8, <u>1158</u>	\FRpm	4
\Deltarxn	5, 8, <u>1161</u>	\FRs	4
\Deltasub	5, 8, <u>1161</u>	\FRt	4
\Deltavap	5, 8, <u>1161</u>	\Fs	4
E		\fsat	5, 7, <u>1150</u>
\EE	4	\fstd	8, <u>1155</u>
		\Ft	4

<code>\residual</code>	9, <u>1390</u>	<code>thermoIntensiveplain (env.)</code>	17, <u>938</u>
S		<code>thermomolesrange (env.)</code>	18, <u>935</u>
<code>\sat</code>	5, 7, <u>1150</u>	<code>thermoNOsubscripts (env.)</code>	18, <u>929</u>
<code>\SE</code>	4	<code>thermoparentheses (env.)</code>	18, <u>903</u>
<code>\SEpm</code>	4	<code>thermoplain (env.)</code>	18, <u>923</u>
<code>\SEs</code>	4	<code>thermosubscripts (env.)</code>	18, <u>932</u>
<code>\SEt</code>	4	<code>thermovmatrix (env.)</code>	<u>1418</u>
<code>\Sm</code>	4	U	
<code>\Spm</code>	4	<code>\UE</code>	4, 9
<code>\SR</code>	4	<code>\UEpm</code>	4, 9
<code>\SRpm</code>	4	<code>\UES</code>	4, 9
<code>\SRs</code>	4	<code>\UET</code>	4, 9
<code>\SRt</code>	4	<code>\Um</code>	3, 4
<code>\Ss</code>	4	<code>\Upm</code>	3, 4
<code>\St</code>	4	<code>\UR</code>	4, 9
<code>\std</code>	8, <u>1155</u>	<code>\URpm</code>	4, 9
<code>\sublimation</code>	5, <u>1161</u>	<code>\URs</code>	4, 9
<code>\sumall</code>	14, <u>1392</u>	<code>\URt</code>	4, 9
<code>\sumallbutlast</code>	14	<code>\Us</code>	4
<code>\summallbutlast</code>	<u>1392</u>	<code>\Ut</code>	3, 4
T		V	
<code>thermobar (env.)</code>	18, <u>918</u>	<code>\vaporization</code>	5, <u>1161</u>
<code>thermobraces (env.)</code>	<u>913</u>	<code>\Vpm</code>	3
<code>thermobrackets (env.)</code>	18, <u>908</u>	W	
<code>thermoextensiveplain (env.)</code>	17, <u>942</u>	<code>\Wm</code>	4
<code>thermoextensivesuperscript (env.)</code>	17, <u>958</u>	<code>\Ws</code>	4
<code>thermoIntensivelowercase (env.)</code>	17, <u>945</u>	<code>\Wt</code>	4