The thermodynamics package*

Karl D. Hammond
hammondkd@missouri.edu

2023/11/16

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:

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

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

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

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

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

*This document corresponds to thermodynamics v2.00, dated 2023/11/16.

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

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

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 $n_1, \ldots, n_C$, 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

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

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 shown in Table 1. Examples using the enthalpy are

\Ht \Hm \Hs \Hpm

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

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

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

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

yields

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

2.2 Partial Molar Properties

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

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

will typeset as

$$\overline{U_i} \quad \overline{U_i^{\mathrm{IG}}} \quad \overline{V_i} \quad \overline{V_i^{\mathrm{IG}}}.$$

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 $\underline{A}$).

					Excess				Residual (Departure)			
Property	Total	Molar	Specific	Partial Molar	T	M	S	PM	T	M	S	PM
Heat	`\Qt`	`\Qm`	`\Qs`									
Work	`\Wt`	`\Wm`	`\Ws`									
Total energy	`\Et`	`\Em`	`\Es`	`\Epm`	`\EEt`	`\EE`	`\EEs`	`\EEpm`	`\ERt`	`\ER`	`\ERs`	`\ERpm`
Internal energy	`\Ut`	`\Um`	`\Us`	`\Upm`	`\UEt`	`\UE`	`\UEs`	`\UEpm`	`\URt`	`\UR`	`\URs`	`\URpm`
Enthalpy	`\Ht`	`\Hm`	`\Hs`	`\Hpm`	`\HEt`	`\HE`	`\HEs`	`\HEpm`	`\HRt`	`\HR`	`\HRs`	`\HRpm`
Entropy	`\St`	`\Sm`	`\Ss`	`\Spm`	`\SEt`	`\SE`	`\SEs`	`\SEpm`	`\SRt`	`\SR`	`\SRs`	`\SRpm`
Volume	`\Vt`	`\Vm`	`\Vs`	`\Vpm`	`\VEt`	`\VE`	`\VEs`	`\VEpm`	`\VRt`	`\VR`	`\VRs`	`\VRpm`
Helmholtz free energy	`\Ft`	`\Fm`	`\Fs`	`\Fpm`	`\FEt`	`\FE`	`\FEs`	`\FEpm`	`\FRt`	`\FR`	`\FRs`	`\FRpm`
Gibbs free energy	`\Gt`	`\Gm`	`\Gs`	`\Gpm`	`\GEt`	`\GE`	`\GEs`	`\GEpm`	`\GRt`	`\GR`	`\GRs`	`\GRpm`
Surface area	`\At`	`\Am`	`\As`	`\Apm`								
Grand potential[a]	`\Lt`	`\Lm`	`\Ls`	`\Lpm`	`\LEt`	`\LE`	`\LEs`	`\LEpm`	`\LRt`	`\LR`	`\LRs`	`\LRpm`
Moles	`\Nt`											

[a] The grand potential, $\underline{\Omega}(T, \underline{V}, \mu_1, \ldots, \mu_C) = \underline{U} - T\underline{S} - \sum_{i=1}^{C} \mu_i n_i$, 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 `\Gpm` 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` `\NewResidualProperty` 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, `\cP_i` renders as $C_{P,i}$, as expected, and `\cP_i^\std` renders as $C^\circ_{P,i}$, also as expected. You can also reverse it: `\cP^\std_i` becomes $C^\circ_{P,i}$.

Name	Macro	Sym.	Definition	Base Symbol Macro
Isobaric heat capacity	`\cP`[a]	cP	$T\left(\frac{\partial S}{\partial T}\right)_P$	`\heatcapacitysymbol`
Isochoric heat capacity	`\cV`[a]	C_V	$T\left(\frac{\partial S}{\partial T}\right)_V$	`\heatcapacitysymbol`
Isothermal compressibility	`\kappaT`	κ_T	$-\frac{1}{V}\left(\frac{\partial V}{\partial P}\right)_T$	`\compressibilitysymbol`
Isentropic compressibility	`\kappaS`	κ_S	$-\frac{1}{V}\left(\frac{\partial V}{\partial P}\right)_S$	`\compressibilitysymbol`
Isobaric expansivity	`\alphaP`	α_P	$\frac{1}{V}\left(\frac{\partial V}{\partial T}\right)_P$	`\expansivitysymbol`
Isentropic expansivity	`\alphaS`	α_S	$\frac{1}{V}\left(\frac{\partial V}{\partial T}\right)_S$	`\expansivitysymbol`
Joule–Thomson coeff.	`\muJT`	μ_{JT}	$\left(\frac{\partial T}{\partial P}\right)_H$	`\JTsymbol`
Pure fugacity	`\fpure`	f	ϕP	
Mixture fugacity	`\fmix`	$\hat{f}$	$\hat{f}_i = x_i\hat{\phi}_i P$	
Saturation fugacity	`\fsat`	f^{sat}	$\phi^{\mathrm{sat}} P^{\mathrm{sat}}$	`\sat`
Pure fugacity coefficient	`\phipure`	ϕ	$\phi_i = \exp\left(\frac{1}{RT}\int_0^P V_i(T,p) - \frac{RT}{p}\,dp\right)$	
Mixture fugacity coeff.	`\phimix`	$\hat{\phi}$	$\hat{\phi}_i = \exp\left(\frac{1}{RT}\int_0^P \overline{V}_i(T,p,\vec{x}) - \frac{RT}{p}\,dp\right)$	
Henry's constant (rational)	`\Henryrat`	h	$\gamma_i^\infty f_i$	
Henry's constant (molal)	`\Henrymol`	$\mathcal{H}$	$M_s\gamma_i^\infty f_i$	
Rational activity coeff.	`\gammarat`	γ^*	γ/γ^∞	
Molal activity coeff.	`\gammamol`	$\gamma^\square$	$x_s\gamma/\gamma^\infty$	
Saturation fugacity coeff.	`\phisat`	ϕ^{sat}	$\phi(T, P^{\mathrm{sat}})$	`\sat`
Saturation pressure	`\Psat`	P^{sat}		`\sat`
Vapor pressure	`\Pvap`	Currently a synonym for `\Psat`[b]		
Standard state	`\std`[c]	$\circ$		
Standard pressure	`\Pstd`	P°		`\std`
Standard fugacity	`\fstd`	f°	$f(T, P^\circ)$	`\std`
Change on mixing	`\Deltamix`[d]	ΔM_{mix}	$M - \sum_{i=1}^C x_i M_i$	`\mixing`
Change on reaction	`\Deltarxn`[d]	ΔM_{rxn}	$\sum_{i=1}^C \nu_i M_i$	`\reaction`
Change on melting	`\Deltafus`[d]	ΔM^{fus}	$M^L - M^S$	`\fusion`
Change on boiling	`\Deltavap`[d]	ΔM^{vap}	$M^V - M^L$	`\vaporization`
Change on subliming	`\Deltasub`[d]	ΔM^{sub}	$M^V - M^S$	`\sublimation`

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

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

[c] Typical usage would be `$\mu_i = \mu^\circ_i + RT \log a_i$`, yielding $\mu_i = \mu_i^\circ + RT\log a_i$.

[d] The usual usage would be something like `$\Deltamix\Vm^\IGM = 0$`.

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, $\underline{B}, B, \hat{B}, \overline{B_i}, \underline{B}^E, B^E, \hat{B}^E, \overline{B_i^E}, \underline{B}^R, B^R, \hat{B}^R$, and $\overline{B_i^R}$ 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` `\cV` `\kappaS` `\kappaT` `\alphaP` `\alphaS`

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 be 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` `\fmix` `\phipure` `\phimix`

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:

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

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` `\gammamol` `\Henryrat` `\Henrymol`

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

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

produces

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

using the default options. These symbols can be customized either directly or by using package options. For example, the `TesterModell` 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` `\Pvap` `\fsat` `\phisat` `\sat`

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

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

\std \Pstd \fstd

The symbol ∘ (\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^\std for convenience to denote standard pressures, and \fstd is defined for standard fugacities so as to ease implementation across textbooks.

The usual usage would be something like this:

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

which produces

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

with the default settings.

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

\Deltamix

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

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

which yields, using the default options,

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

\mixing

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

\Deltafus \Deltasub \Deltavap

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

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

yielding

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

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.

\Deltaf \Deltarxn

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

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

results in

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

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
\URt
\URs
\URpm
\UE
\UEt
\UEs
\UEpm
\excess
\residual

Additional macros are defined that make it easy to typeset the residual (also called "departure") and excess total, molar, specific, and partial molar properties. These macros follow the same pattern: `\UR`, `\URt`, `\URs`, and `\URpm` typeset the molar, total, specific, and partial molar residual internal energies, respectively, and by default 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, `\UE`, `\UEt`, `\UEs`, and `\UEpm` typeset the corresponding excess properties. The first character of the macros for other properties follow the same pattern as in Table 1.

The R and E characters are generated by the macros `\residual` and `\excess`, respectively. These macros can be redefined; for example, if you want `\SE`, which 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

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

using the default options.

2.6 Partial Derivatives

\Partial
\Partial*

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

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

which yields

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

\Partialinline

Inline first derivatives[2] 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
\PartialSecond*
\PartialMixSecond
\PartialMixSecond*

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

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

looks like

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

\PartialSecondinline
\PartialMixSecondinline

Inline versions[2] 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}
```

[1] You knew that joke was coming.

[2] The "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

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

2.6.2 Delimiter Sizing

\PartialBigg
\PartialSecondBigg
\PartialMixSecondBigg
\Partialbigg
\PartialSecondbigg
\PartialMixSecondbigg

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

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

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

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

produces

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

\Partialinlinetext
\PartialSecondinlinetext
\PartialMixSecondinlinetext

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

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

2.6.3 Higher-Order Derivatives

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

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

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`
`\allNsbut`
`\allmus`
`\allmusbut`
`\allXs`
`\allXsbut`
`\allYs`
`\allYsbut`
`\allMs`
`\allMsbut`
`\allWs`
`\allWsbut`

It is common in thermodynamics to use notation such as

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

or perhaps

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

or

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

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

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

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 $n_1, \ldots, n_C$ 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,

```
\[ \Partial{\Ht}{\Nt_1}{T,P,\allNsbut{1}} =
   \Partial{\Ht}{\Nt_1}{T,P,\allNsbut[m]{1}} = \Hpm_1 \]
```

typesets as

$$\left(\frac{\partial \overline{H}}{\partial n_1}\right)_{T,P,n_{j\neq 1}} = \left(\frac{\partial \overline{H}}{\partial n_1}\right)_{T,P,n_{m\neq 1}} = \overline{H_1}.$$

Similarly,

```
\[ \Partial{\Hm}{x_i}{T,P,\allXsbut{i}}
  = \Partial{\Hm}{x_i}{T,P,\allXsbut[m]{i}}
  = \Hpm_i - \Hpm_\ncomponents \]
```

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 $x_1, \ldots, [x_k], \ldots, x_{C-1}$ 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` `\allbutlastand`

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

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

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

2.8 Jacobians

`\Jacobian` `\Jacobiandet`

The Jacobian determinant is often denoted with Leibnitz-like notation, viz.,

```
\[ \Jacobian{K,L}{X,Y} = \Jacobiandet{K,L}{X,Y} \],
```

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 `\Jacobiandet`. 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.,

```
\[ \Jacobian{K,L,M}{X,Y,Z}
  = \Jacobiandet[\displaystyle][3ex]{K,L,M}{X,Y,Z} \]
```

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

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

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

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

2.9 Sums and Products

`\sumall` `\sumallbutlast` `\prodall`

It is common to require sums and products such as

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

This package defines shortcuts to typeset such terms thus:

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

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	`\Et`	`\Ut`	`\Ft`	`\Gt`	`\Ht`	`\At`	`\Nt`
`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:

```
\[ \Ft = \Ut - T\St = -P\Vt + \sumall_i \mu_i \Nt_i + \sigma \At
   \qquad \Hm = \Um + P\Vm \qquad \Et = \Ut + \frac12 mv^2 \],
```

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

Option	`\Vt`	`\Vm`	`\Vs`	`\Vpm_i`
`intensive-plain`	$\underline{V}$	V	$\hat{V}$	$\overline{V}_i$
`extensive-plain`	V	$\underline{V}$	$\hat{V}$	$\overline{V}_i$
`extensive-superscript`	V^t	V	$\hat{V}$	$\overline{V}_i$
`intensive-lowercase`	V	v	$\hat{v}$	$\overline{v}_i$

which would look like

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

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

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

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 $\overline{V}_j$, 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:

$$\overline{H}_i = \left(\frac{\partial \underline{H}}{\partial n_i}\right)_{T,P,n_{j\neq i}} = \left(\frac{\partial nH}{\partial n_i}\right)_{T,P,n_{j\neq i}} \qquad \texttt{intensive-plain}$$

$$\overline{H}_i = \left(\frac{\partial H}{\partial n_i}\right)_{T,P,n_{j\neq i}} = \left(\frac{\partial n\underline{H}}{\partial n_i}\right)_{T,P,n_{j\neq i}} \qquad \texttt{extensive-plain}$$

$$\overline{H}_i = \left(\frac{\partial H^t}{\partial n_i}\right)_{T,P,n_{j\neq i}} = \left(\frac{\partial nH}{\partial n_i}\right)_{T,P,n_{j\neq i}} \qquad \texttt{extensive-superscript}$$

$$\overline{h}_i = \left(\frac{\partial H}{\partial n_i}\right)_{T,P,n_{j\neq i}} = \left(\frac{\partial nh}{\partial n_i}\right)_{T,P,n_{j\neq i}} \qquad \texttt{intensive-lowercase}$$

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.

thermointensiveplain (*env.*)
thermoextensiveplain (*env.*)
thermoextensivesuperscript (*env.*)
thermointensivelowercase (*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 `thermointensiveplain` 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`, `\St` will become S^t and `\Sm` will be S; and inside `thermointensivesubscript`, `\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

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

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

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

and instead renders them

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

Combined with `plain-derivatives`, this would give

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

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.*) `thermobrackets` (*env.*) `thermobar` (*env.*) `thermoplain` (*env.*) `thermosubscripts` (*env.*) `thermoNOsubscripts` (*env.*)

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

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

3.1.4 Options Regarding the Number of Moles

`\allNs` `\allNsbut`

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

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

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

You can change these to expand to ranges using the `moles-range` option, which renders `\allNs` as $n_1,\ldots,n_C$ and `\allNsbut{i}` as $n_1,\ldots,[n_i],\ldots,n_C$. 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 đ. 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`
`\allNs`	$\vec{n}$	$n_1,\ldots,n_C$
`\allNsbut{1}`	$n_{j\neq 1}$	$n_2,\ldots,n_C$
`\allNsbut{i}`	$n_{j\neq i}$	$n_1,\ldots,[n_i],\ldots,n_C$
`\allNsbut{j}`	$n_{k\neq j}$	$n_1,\ldots,[n_j],\ldots,n_C$
`\allNsbut{\ncomponents}`	$n_{j\neq C}$	$n_1,\ldots,n_{C-1}$
`\allNsbut[k]{i}`	$n_{k\neq i}$	$n_1,\ldots,[n_i],\ldots,n_C$
`\allXs`	$\vec{x}$	$x_1,\ldots,x_C$
`\allXsbut{1}`	$x_{j\neq 1,C}$	$x_2,\ldots,x_{C-1}$
`\allXsbut{i}`	$x_{j\neq i,C}$	$x_1,\ldots,[x_i],\ldots,x_{C-1}$
`\allXsbut{j}`	$x_{k\neq j,C}$	$x_1,\ldots,[x_j],\ldots,x_{C-1}$
`\allXsbut{\ncomponents-1}`	$x_{j\neq C-1,C}$	$x_1,\ldots,x_{C-2}$
`\allXsbut[k]{\ncomponents-1}`	$x_{k\neq C-1,C}$	$x_1,\ldots,x_{C-2}$
`\allXsbut{\ncomponents}`[b]	$x_{j\neq C}$	$x_1,\ldots,x_{C-1}$

[a] You 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.

[b] This would typically be used to denote something like $\underline{G}(T,P,n,x_1,\ldots,x_{C-1})$ 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, \expansitivitysymbol, \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, \phipure, \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, \phipure,

\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.

```
\ExplSyntaxOn
\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.

```
\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.

```
\AtBeginDocument{
    \@ifpackageloaded{pxfonts}{%
        \ProvideDocumentCommand{\dbar}{}
            {\mkern5mu\mathchar'26\mkern-10mu d}
    }{}
    \@ifpackageloaded{newpxmath}{%
        \ProvideDocumentCommand{\dbar}{}
            {\mkern5mu\mathchar'26\mkern-10mu d}
    }{}
    \@ifpackageloaded{txfonts}{%
        \ProvideDocumentCommand{\dbar}{}
            {\mkern5mu\mathchar'26\mkern-11mu d}
    }{}
    \@ifpackageloaded{mathptmx}{%
        \ProvideDocumentCommand{\dbar}{}
            {\mkern5mu\mathchar'26\mkern-10mu d}
    }{}
```

```
    \@ifpackageloaded{newtxmath}{%
        \ProvideDocumentCommand{\dbar}{}
            {\mkern5mu\mathchar'26\mkern-11mu d}
    }{}
    \@ifpackageloaded{mathdesign}{%
        \tl_const:Nn \c_@@_charter_tl {mdbch}
        \tl_const:Nn \c_@@_utopia_tl {mdput}
        \tl_const:Nn \c_@@_garamond_tl {mdugm}
        \tl_if_eq:NNT \MD@default@family \c_@@_utopia_tl
        { \ProvideDocumentCommand{\dbar}{}
            {\mkern5mu\mathchar'26\mkern-20mu d}
        }
        \tl_if_eq:NNT \MD@default@family \c_@@_charter_tl
        { \ProvideDocumentCommand{\dbar}{}
            {\mkern5mu\mathchar'26\mkern-15mu d}
        }
        \tl_if_eq:NNT \MD@default@family \c_@@_garamond_tl
        { \ProvideDocumentCommand{\dbar}{}
            {\mkern5mu\mathchar'26\mkern-17mu d}
        }
    }{}
    % Defaults to Computer Modern
    \ProvideDocumentCommand{\dbar}{}
        {\mkern3mu\mathchar'26\mkern-12mu d}
}
```

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

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, expansivitites, 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.

```
\tl_new:N \g_@@_total_energy_symbol
\tl_new:N \g_@@_internal_energy_symbol
\tl_new:N \g_@@_Helmholtz_symbol
\tl_new:N \g_@@_Gibbs_symbol
\tl_new:N \g_@@_Landau_symbol
\tl_new:N \g_@@_enthalpy_symbol
\tl_new:N \g_@@_entropy_symbol
\tl_new:N \g_@@_area_symbol
\tl_new:N \g_@@_volume_symbol
```

```
\tl_new:N \g_@@_mole_symbol
\tl_new:N \g_@@_heat_symbol
\tl_new:N \g_@@_work_symbol
\tl_new:N \g_@@_temperature_symbol
\tl_new:N \g_@@_pressure_symbol

\tl_gset:Nn \g_@@_total_energy_symbol E
\tl_gset:Nn \g_@@_internal_energy_symbol U
\tl_gset:Nn \g_@@_Helmholtz_symbol A
\tl_gset:Nn \g_@@_Gibbs_symbol G
\tl_gset:Nn \g_@@_Landau_symbol \Omega
\tl_gset:Nn \g_@@_enthalpy_symbol H
\tl_gset:Nn \g_@@_entropy_symbol S
\tl_gset:Nn \g_@@_area_symbol a
\tl_gset:Nn \g_@@_volume_symbol V
\tl_gset:Nn \g_@@_mole_symbol n
\tl_gset:Nn \g_@@_heat_symbol Q
\tl_gset:Nn \g_@@_work_symbol W
\tl_gset:Nn \g_@@_temperature_symbol T
\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.

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

`\PartialOpen` `\PartialClose` `\PartialEmptyClose`

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`, 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.

```
\tl_new:N \l_@@_PartialOpen_tl
\tl_new:N \l_@@_PartialEmptyClose_tl
\tl_new:N \l_@@_PartialClose_tl

\tl_set:Nn \l_@@_PartialOpen_tl {(}
\tl_set:Nn \l_@@_PartialClose_tl {)}
\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.

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

```
\DeclareOption{EUHAGaN}{\ExecuteOptions{EUAGHaN}}
\DeclareOption{EUFGHAn}{%
  \tl_gset:Nn \g_@@_Helmholtz_symbol F
  \tl_gset:Nn \g_@@_area_symbol A
}
\DeclareOption{EUFGHAN}{%
  \tl_gset:Nn \g_@@_Helmholtz_symbol F
  \tl_gset:Nn \g_@@_area_symbol A
  \tl_gset:Nn \g_@@_mole_symbol N
}
\DeclareOption{EEFGHAn}{%
  \tl_gset:Nn \g_@@_total_energy_symbol {\mathcal{E}}
  \tl_gset:Nn \g_@@_internal_energy_symbol E
  \tl_gset:Nn \g_@@_Helmholtz_symbol F
  \tl_gset:Nn \g_@@_area_symbol A
}
\DeclareOption{EEFGHAN}{%
  \tl_gset:Nn \g_@@_total_energy_symbol {\mathcal{E}}
  \tl_gset:Nn \g_@@_internal_energy_symbol E
  \tl_gset:Nn \g_@@_Helmholtz_symbol F
  \tl_gset:Nn \g_@@_area_symbol A
  \tl_gset:Nn \g_@@_mole_symbol N
}
\DeclareOption{EEFGHan}{%
  \tl_gset:Nn \g_@@_total_energy_symbol {\mathcal{E}}
  \tl_gset:Nn \g_@@_internal_energy_symbol E
  \tl_gset:Nn \g_@@_Helmholtz_symbol F
}
\DeclareOption{EEFGHaN}{%
  \tl_gset:Nn \g_@@_total_energy_symbol {\mathcal{E}}
  \tl_gset:Nn \g_@@_internal_energy_symbol E
  \tl_gset:Nn \g_@@_Helmholtz_symbol F
  \tl_gset:Nn \g_@@_mole_symbol N
}
\DeclareOption{EEAGHan}{%
  \tl_gset:Nn \g_@@_total_energy_symbol {\mathcal{E}}
  \tl_gset:Nn \g_@@_internal_energy_symbol E
}
\DeclareOption{EEAGHaN}{%
  \tl_gset:Nn \g_@@_total_energy_symbol {\mathcal{E}}
  \tl_gset:Nn \g_@@_internal_energy_symbol E
  \tl_gset:Nn \g_@@_mole_symbol N
}
\DeclareOption{EUAGHAn}{%
  \tl_gset:Nn \g_@@_area_symbol {\mathcal{A}}
}
\DeclareOption{EUAGHAN}{%
  \tl_gset:Nn \g_@@_area_symbol {\mathcal{A}}
  \tl_gset:Nn \g_@@_mole_symbol N
}
\DeclareOption{EUFGHan}{%
  \tl_gset:Nn \g_@@_Helmholtz_symbol F
}
\DeclareOption{EUFGHaN}{%
  \tl_gset:Nn \g_@@_Helmholtz_symbol F
  \tl_gset:Nn \g_@@_mole_symbol N
```

```
}
```

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

```
\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.

```
\cs_new:Nn \@@_extensive:n {#1}
\cs_new:Nn \@@_intensive:n {#1}
\cs_new:Nn \@@_specific:n {\hat{#1}}

\cs_new:Npn \@@_set_intensive_plain
{
  \cs_set:Nn \@@_extensive:n {\@@_underline:n{##1}}
  \cs_set:Nn \@@_intensive:n {##1}
}
\cs_new:Npn \@@_set_extensive_plain
{
  \cs_set:Nn \@@_extensive:n {##1}
  \cs_set:Nn \@@_intensive:n {\@@_underline:n{##1}}
}
\cs_new:Npn \@@_set_lowercase_pms
{
  \RenewDocumentCommand{\partialmolar}{m}
  {
    \tl_set:Nn \l_@@_pm_symbol_tl {\text_lowercase:n {##1}}
    \@@_generic_pm:
  }
}
\cs_new:Npn \@@_set_intensive_lowercase
{
  \cs_set:Nn \@@_extensive:n {\text_uppercase:n {##1}}
  \cs_set:Nn \@@_intensive:n {\text_lowercase:n {##1}}
  \cs_set:Nn \@@_specific:n {\hat{\text_lowercase:n {##1}}}
}
\cs_new:Npn \@@_set_extensive_superscripts
{
  \cs_set:Nn \@@_extensive:n {{##1}\c_math_superscript_token t}
  \cs_set:Nn \@@_intensive:n {##1}
}
\DeclareOption{extensive-plain}{\@@_set_extensive_plain}
\DeclareOption{intensive-plain}{\@@_set_intensive_plain} % the default
\DeclareOption{intensive-lowercase}{% PLEASE don't use this!
  \@@_set_intensive_lowercase
  \AtEndOfPackage{
    \@@_set_lowercase_pms
    \RenewDocumentCommand{\heatcapacitysymbol}{}{c}
  }
}
\DeclareOption{extensive-superscript}{%
  \@@_set_extensive_superscripts
  \AtEndOfPackage{%
    \RenewDocumentCommand{\URt}{}{\g_@@_internal_energy_symbol
      \c_math_superscript_token{\residual,t}}
```

```
    \RenewDocumentCommand{\HRt}{}{\g_@@_enthalpy_symbol
      \c_math_superscript_token{\residual,t}}
    \RenewDocumentCommand{\FRt}{}{\g_@@_Helmoholtz_symbol
      \c_math_superscript_token{\residual,t}}
    \RenewDocumentCommand{\GRt}{}{\g_@@_Gibbs_symbol
      \c_math_superscript_token{\residual,t}}
    \RenewDocumentCommand{\VRt}{}{\g_@@_volume_symbol
      \c_math_superscript_token{\residual,t}}
    \RenewDocumentCommand{\SRt}{}{\g_@@_entropy_symbol
      \c_math_superscript_token{\residual,t}}
    \RenewDocumentCommand{\UEt}{}{\g_@@_internal_energy_symbol
      \c_math_superscript_token{\excess,t}}
    \RenewDocumentCommand{\HEt}{}{\g_@@_enthalpy_symbol
      \c_math_superscript_token{\excess,t}}
    \RenewDocumentCommand{\FEt}{}{\g_@@_Helmoholtz_symbol
      \c_math_superscript_token{\excess,t}}
    \RenewDocumentCommand{\GEt}{}{\g_@@_Gibbs_symbol
      \c_math_superscript_token{\excess,t}}
    \RenewDocumentCommand{\VEt}{}{\g_@@_volume_symbol
      \c_math_superscript_token{\excess,t}}
    \RenewDocumentCommand{\SEt}{}{\g_@@_entropy_symbol
      \c_math_superscript_token{\excess,t}}
  }
}
```

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.

```
\bool_new:N \l_@@_subscripted_bool
\bool_set_true:N \l_@@_subscripted_bool
\DeclareOption{subscripts}{\bool_set_true:N \l_@@_subscripted_bool}
\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.

```
\DeclareOption{parentheses}{}
\DeclareOption{brackets}{%
  \tl_set:Nn \l_@@_PartialOpen_tl {[}
  \tl_set:Nn \l_@@_PartialClose_tl {]}
  \tl_set:Nn \l_@@_PartialEmptyClose_tl {]}
}
\DeclareOption{braces}{%
  \tl_set:Nn \l_@@_PartialOpen_t1 {\{}
  \tl_set:Nn \l_@@_PartialClose_t1 {\}}
  \tl_set:Nn \l_@@_PartialEmptyClose_tl {\}}
}
\DeclareOption{bar}{%
  \tl_set:Nn \l_@@_PartialOpen_tl {.}
  \tl_set:Nn \l_@@_PartialClose_tl {\rvert}
  \tl_set:Nn \l_@@_PartialEmptyClose_tl {.}
}
\DeclareOption{plain-derivatives}{% This implies dU(S,V,N)/dS notation
```

```
  \tl_set:Nn \l_@@_PartialOpen_tl {.}
  \tl_set:Nn \l_@@_PartialClose_tl {.}
  \tl_set:Nn \l_@@_PartialEmptyClose_tl {.}
  \ExecuteOptions{nosubscripts}
}
```

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.

```
\NewExpandableDocumentCommand \ncomponents {} {C}
```

\allNs \allXs \allYs \allmus \allMs \allWs Several macros define a shorthand for "moles of all species" (\allNs) and "moles of all species except" (\allNsbut), as well as similar quantities for masses (\allMs, \allMsbut) and chemical potentials (\allmus, \allmusbut), which occur frequently in mixture thermodynamics. The default is for \allNs to become $\vec{n}$ and allNsbut{i} to become $n_{j\neq i}$. The optional argument changes which index (default: j) to use in the left side of the inequality.[3] Essentially identical commands are defined for chemical potentials and masses: \allmus and \allmusbut and \allMs and \allMsbut, respectively.

```
\NewDocumentCommand{\allNs}{O{i}}{\allcomponents[#1]{\Nt}}
\NewDocumentCommand{\allXs}{O{i}}{\allcomponents[#1]{x}}
\NewDocumentCommand{\allYs}{O{i}}{\allcomponents[#1]{y}}
\NewDocumentCommand{\allmus}{O{i}}{\allcomponents[#1]{\mu}}
\NewDocumentCommand{\allMs}{O{i}}{\allcomponents[#1]{m}}
\NewDocumentCommand{\allWs}{O{i}}{\allcomponents[#1]{w}}
```

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

```
\[ \Partial{\Gm}{T}{P,\allXs} = -\Sm \qquad
   \Partial{\Gm}{x_i}{T,P,\allXsbut{i}} \neq \Gpm_i \]
```

yields

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

```
\NewDocumentCommand{\allNsbut}{O{j} m} {\allbut[#1]{#2}{\Nt}}
\NewDocumentCommand{\allXsbut}{O{j} m} {\allbutlastand[#1]{#2}{x}}
\NewDocumentCommand{\allYsbut}{O{j} m} {\allbutlastand[#1]{#2}{y}}
\NewDocumentCommand{\allmusbut}{O{j} m} {\allbut[#1]{#2}{\mu}}
\NewDocumentCommand{\allMsbut}{O{j} m} {\allbut[#1]{#2}{m}}
\NewDocumentCommand{\allWsbut}{O{j} m} {\allbutlastand[#1]{#2}{w}}
```

\allbutlastand \allbut \allcomponents 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

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

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

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

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

```
\NewDocumentCommand{\allcomponents}{O{} m}{\vec{#2}}
\NewDocumentCommand{\allbut}{O{j} m m}
{ \tl_if_eq:nnTF {#1} {#2}
  { {#3}\c_math_subscript_token{k \neq #2} }
  { {#3}\c_math_subscript_token{#1 \neq #2} }
}
\NewDocumentCommand{\allbutlastand}{O{j} m m}
{ \tl_if_eq:xxTF {#2} {\ncomponents}
  { {#3}\c_math_subscript_token{#1 \neq #2} }
  { \tl_if_eq:nnTF {#1} {#2}
    { {#3}\c_math_subscript_token{k \neq #2,\ncomponents} }
    { {#3}\c_math_subscript_token{#1 \neq #2,\ncomponents} }
  }
}
```

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

```
\DeclareOption{moles-index}{}
\DeclareOption{moles-range}{ \@@_set_moles_range }
\cs_new:Npn \@@_set_moles_range {%
  \RenewDocumentCommand{\allcomponents}{O{} m}
  { {##2}\c_math_subscript_token 1,\dots,
        {##2}\c_math_subscript_token{\ncomponents} }
  \RenewDocumentCommand{\allbut}{O{j} m m}
  { \tl_if_eq:nnTF {##2} {1}
    { {##3}\c_math_subscript_token 2,\dots,
        {##3}\c_math_subscript_token{\ncomponents} }
    { \tl_if_eq:xxTF {##2} {\ncomponents}
      { {##3}\c_math_subscript_token 1,\dots,
          {##3}\c_math_subscript_token{\ncomponents-1} }
      { {##3}\c_math_subscript_token 1,\dots,
            [{##3}\c_math_subscript_token{##2}],
            \dots,{##3}\c_math_subscript_token{\ncomponents} }
    }
  }
  \RenewDocumentCommand{\allbutlastand}{O{j} m m}
  { \tl_if_eq:nnTF {##2} {1}
    { {##3}\c_math_subscript_token 2,\dots,
        {##3}\c_math_subscript_token{\ncomponents-1} }
    { \tl_if_eq:xxTF {##2} {\ncomponents}
      { {##3}\c_math_subscript_token 1,\dots,
        {##3}\c_math_subscript_token{\ncomponents-1}
      }
      { \tl_if_eq:xxTF {##2} {\ncomponents-1}
        { {##3}\c_math_subscript_token 1,\dots,
          {##3}\c_math_subscript_token{\ncomponents-2} }
        {
          {##3}\c_math_subscript_token 1,\dots,
          [{##3}\c_math_subscript_token{##2}],\dots,
           {##3}\c_math_subscript_token{\ncomponents-1}
        }
      }
    }
  }
}
```

The remaining options define textbook-specific notation.

```
\DeclareOption{Bejan}{
```

```
  \ExecuteOptions{EUFGHAN,intensive-lowercase,delta}
  \cs_set:Nn \@@_specific:n {\text_lowercase:n {#1}}
  \cs_set:Nn \@@_intensive:n {\bar{\text_lowercase:n {#1}}}
  \tl_gset:Nn \g_@@_volume_symbol v
}
\DeclareOption{CBK}{
  \ExecuteOptions{EUAGHAN,intensive-lowercase}
  \AtEndOfPackage{
    \cs_set:Nn \@@_overline:n {\widetilde{#1}}
    \cs_set:Nn \@@_specific:n {\text_lowercase:n {#1}}
    \cs_set:Nn \@@_intensive:n {\bar{\text_lowercase:n{#1}}}
    \tl_gset:Nn \g_@@_pressure_symbol p
    \RenewDocumentCommand{\Deltarxn}{m}{{#1}\c_math_subscript_token R}
    \RenewDocumentCommand{\compressibilitysymbol}{}{\beta}
    \RenewDocumentCommand{\expansivitysymbol}{}{\alpha}
  }
}
\DeclareOption{ElliottLira}{
  \AtEndOfPackage{
    \RenewDocumentCommand{\allcomponents}{O{} m}{#2}
    \RenewDocumentCommand{\Deltarxn}{m}{\Delta #1}
    \RenewDocumentCommand{\IG}{}{{\text{ig}}}
    \RenewDocumentCommand{\IGM}{}{{\text{ig}}}
    \RenewDocumentCommand{\IS}{}{{\text{is}}}
  }
}
\DeclareOption{Koretsky}{
  \ExecuteOptions{EUAGHAn,brackets,intensive-lowercase,delta}
  \AtEndOfPackage{
    %^^A Undo part of intensive-lowercase
    \RenewDocumentCommand{\partialmolar}{m}
    { \tl_set:Nn \l_@@_pm_symbol_tl {#1}
      \@@_generic_pm:
    }
    \RenewDocumentCommand{\expansivitysymbol}{}{\beta}
    \RenewDocumentCommand{\IS}{}{{\text{ideal}}}
    \RenewDocumentCommand{\residual}{}{{\text{dep}}}
    \RenewDocumentCommand{\IG}{}{{\text{ideal}}}
    \RenewDocumentCommand{\IGM}{}{{\text{ideal}}}
    \RenewDocumentCommand{\Henryrat}{}{{\mathcal{H}}}
    \RenewDocumentCommand{\gammarat}{}
        {\gamma\c_math_superscript_token\text{Henry's}}
    \RenewDocumentCommand{\phipure}{}{\varphi}
    \RenewDocumentCommand{\phimix}{}{\hat\varphi}
    \RenewDocumentCommand{\phisat}{}{\varphi\c_math_superscript_token\sat}
    \cs_new:Npn \Delta_fus_sym {} {}
    \NewSubscriptedSymbol{\Delta_fus}{\Delta_fus_sym}{\fusion}
    \RenewDocumentCommand{\Deltafus}{m}{
      \cs_set:Npn \Delta_fus_sym {} { \Delta #1 }
      \Delta_fus
    }
    \cs_new:Npn \Delta_vap_sym {} {}
    \NewSubscriptedSymbol{\Delta_vap}{\Delta_vap_sym}{\vaporization}
    \RenewDocumentCommand{\Deltavap}{m}{
      \cs_set:Npn \Delta_vap_sym {} { \Delta #1 }
      \Delta_vap
```

```
    }
    \cs_new:Npn \Delta_sub_sym {} {}
    \NewSubscriptedSymbol{\Delta_sub}{\Delta_sub_sym}{\sublimation}
    %^^A Undo part of intensive-lowercase
    \RenewDocumentCommand{\Deltasub}{m}{
      \cs_set:Npn \Delta_sub_sym {} { \Delta #1 }
      \Delta_sub
    }
  }
}
\DeclareOption{MSBB}{
  \ExecuteOptions{EUFGHAn,intensive-lowercase,delta}
  \AtEndOfPackage{
    \RenewDocumentCommand{\IGM}{}{\ast}
    \RenewDocumentCommand{\IG}{}{\ast}
    \RenewDocumentCommand{\expansivitysymbol}{}{\beta}
    \RenewDocumentCommand{\allcomponents}{O{} m}{#2}
    \RenewDocumentCommand{\allbut}{O{j} m m}
    {
      \tl_if_eq:nnTF {#1} {#2}
      { {#3}\c_math_subscript_token k }
      { {#3}\c_math_subscript_token{#1} }
    }
    \tl_gset_eq:NN \g_@@_Helmholtz_symbol \psi
    \RenewDocumentCommand{\Ft}{}{\Psi}
    \cs_set:Nn \@@_intensive:n {\@@_overline:n{\text_lowercase:n{#1}}}
    \cs_set:Nn \@@_specific:n {\text_lowercase:n{#1}}
    \RenewDocumentCommand{\fmix}{}{\bar f}
    \RenewDocumentCommand{\phimix}{}{\bar\phi}
    \RenewDocumentCommand{\phimix}{}{\bar\phi}
    \tl_gset:Nn \g_@@_pressure_symbol p
    \RenewDocumentCommand{\partialmolar}{m}
    {
      \tl_set:Nn \l_@@_pm_symbol_tl {#1}
      \@@_generic_pm:
    }
  }
}
\DeclareOption{Prausnitz}{
  \ExecuteOptions{intensive-lowercase}
  \AtEndOfPackage{
    \RenewDocumentCommand{\fmix}{}{f}
    \RenewDocumentCommand{\phimix}{}{\phi}
    \RenewDocumentCommand{\fsat}{}{\fpure\c_math_superscript_token\sat}
    % TODO: this should pick up H_2 and make it into H_{2,1} (assuming the
    % solvent is always 1...?)
    \RenewDocumentCommand{\Henryrat}{}{H}
    \RenewDocumentCommand{\residual}{}{{{\mathcal{R}}}}
    \RenewDocumentCommand{\allcomponents}{O{i} m}
      { {#2}\c_math_subscript_token{#1} }
    \RenewDocumentCommand{\allbut}{O{i} m m}
    { \tl_if_eq:nnTF {#1} {#2}
        { {#3}\c_math_subscript_token k }
        { {#3}\c_math_subscript_token{#1} }
    }
    \RenewSubscriptedSymbol{\fpure}{f}{{\text{pure}}}
```

```
    \RenewSubscriptedSymbol{\phipure}{\phi}{{\text{pure}}}
  }
}
\DeclareOption{Sandler}{
  \ExecuteOptions{EUAGHaN,extensive-plain}
  \AtEndOfPackage{
    \RenewDocumentCommand{\sat}{}{{\text{vap}}}
    \RenewDocumentCommand{\excess}{}{{\text{ex}}}
    \RenewDocumentCommand{\residual}{}{{\text{r}}}
    \RenewExpandableDocumentCommand{\ncomponents}{}{\mathcal{C}}
    \RenewDocumentCommand{\fmix}{}{\bar f}
    \RenewDocumentCommand{\fstd}{}{\bar f\c_math_superscript_token\std}
    \RenewDocumentCommand{\phimix}{}{\bar\phi}
    \RenewDocumentCommand{\allcomponents}{O{} m}{\@@_underline:n{#2}}
    \RenewDocumentCommand{\IG}{}{{\text{IG}}}
    \RenewDocumentCommand{\IGM}{}{{\text{IGM}}}
    \RenewDocumentCommand{\IS}{}{{\text{IM}}}
    \RenewDocumentCommand{\Deltamix}{m}
        {\Delta\c_math_subscript_token\mixing #1}
    \RenewDocumentCommand{\Deltarxn}{m}
        {\Delta\c_math_subscript_token\reaction #1}
    \RenewDocumentCommand{\Deltasub}{m}
        {\Delta\c_math_subscript_token\sublimation #1}
    \RenewDocumentCommand{\Deltafus}{m}
        {\Delta\c_math_subscript_token\fusion #1}
    \RenewDocumentCommand{\Deltavap}{m}
        {\Delta\c_math_subscript_token\vaporization #1}
    \RenewDocumentCommand{\Henryrat}{}{H}
    \RenewSubscriptedSymbol{\cV}
        {\heatcapacitysymbol}{\g_@@_volume_symbol}
    \RenewSubscriptedSymbol{\cP}
        {\heatcapacitysymbol}{\g_@@_pressure_symbol}
    \RenewSubscriptedSymbol{\cVt}
        {\Nt\heatcapacitysymbol}{\g_@@_volume_symbol}
    \RenewSubscriptedSymbol{\cPt}
        {\Nt\heatcapacitysymbol}{\g_@@_pressure_symbol}
  }
}
\DeclareOption{SVNAS}{
  \ExecuteOptions{extensive-superscript}
  \AtEndOfPackage{
    \RenewDocumentCommand{\allcomponents}{O{} m}{#2}
    \RenewDocumentCommand{\allbut}{O{j} m m}
    {
      \tl_if_eq:nnTF {#1} {#2}
      { {#3}\c_math_subscript_token k }
      { {#3}\c_math_subscript_token{#1} }
    }
    \RenewDocumentCommand{\IG}{}{{ig}}
    \RenewDocumentCommand{\IGM}{}{{ig}}
    \RenewDocumentCommand{\IS}{}{{id}}
    \RenewDocumentCommand{\expansivitysymbol}{}{\beta}
    \RenewDocumentCommand{\Deltarxn}{m}{\Delta #1}
    \RenewSubscriptedSymbol{\cVt}
        {\Nt\heatcapacitysymbol}{\g_@@_volume_symbol}
    \RenewSubscriptedSymbol{\cPt}
```

```
        {\Nt\heatcapacitysymbol}{\g_@@_pressure_symbol}
  }
}
\DeclareOption{TesterModell}{
  \ExecuteOptions{EUAGHaN,delta}
  \AtEndOfPackage{
    \RenewExpandableDocumentCommand{\ncomponents}{}{n}
    \RenewDocumentCommand{\allcomponents}{O{i} m}
    {
      {#2}\c_math_subscript_token{#1}
    }
    \RenewDocumentCommand{\allbut}{O{i} m m}
    { \tl_if_eq:nnTF {#1} {#2}
      { {#3}\c_math_subscript_token{k}[#2] }
      { {#3}\c_math_subscript_token{#1}[#2] }
    }
    \RenewDocumentCommand{\allbutlastand}{O{j} m m}
    { \tl_if_eq:xxTF {#2} {\ncomponents}
      { {#3}\c_math_subscript_token{#1}\relax[#2] }
      { \tl_if_eq:nnTF {#1} {#2}
        { {#3}\c_math_subscript_token{k}[#2,\ncomponents] }
        { {#3}\c_math_subscript_token{#1}[#2,\ncomponents] }
      }
    }
    \RenewDocumentCommand{\IG}{}{{o}}
    \RenewDocumentCommand{\IGM}{}{{o}}
    \RenewDocumentCommand{\IS}{}{{ID}}
    \RenewDocumentCommand{\excess}{}{{EX}}
    \RenewDocumentCommand{\reaction}{}{{rx}}
    \RenewDocumentCommand{\Henryrat}{}
        {f\c_math_superscript_token{\ast\ast}}
    \RenewDocumentCommand{\Henrymol}{}{f\c_math_superscript_token\ast}
    \RenewDocumentCommand{\gammarat}{}
        {\gamma\c_math_superscript_token{\ast\ast}}
    \RenewDocumentCommand{\gammamol}{}
        {\gamma\c_math_superscript_token\ast}
    \RenewExpandableDocumentCommand{\JTsymbol}{}{\alpha}
    \RenewSubscriptedSymbol{\muJT}{\JTsymbol}{H}
  }
}
\DeclareOption{Thompson}{
  \ExecuteOptions{EUAGHAn,delta}
  \AtEndOfPackage
  {
    \RenewDocumentCommand{\excess}{}{{EX}}
    \RenewDocumentCommand{\residual}{}{{R}}
    %^^A He uses k_H for both types of Henry's constant, but I changed
    %^^A the rational basis one just so there is SOME difference
    \RenewSubscriptedSymbol{\Henryrat}{k}{h}
    \RenewSubscriptedSymbol{\Henrymol}{k}{H}
    \RenewDocumentCommand{\allcomponents}{O{j} m}
    {
      {#2}\c_math_subscript_token{#1}
    }
    \RenewDocumentCommand{\allNs}{O{j}}{\allcomponents[#1]{\Nt}}
    \RenewDocumentCommand{\allXs}{O{j}}{\allcomponents[#1]{x}}
```

```
\RenewDocumentCommand{\allYs}{O{j}}{\allcomponents[#1]{y}}
\RenewDocumentCommand{\allmus}{O{j}}{\allcomponents[#1]{\mu}}
\RenewDocumentCommand{\allMs}{O{j}}{\allcomponents[#1]{m}}
\RenewDocumentCommand{\allWs}{O{j}}{\allcomponents[#1]{w}}
\RenewExpandableDocumentCommand{\ncomponents}{}{c}
\RenewDocumentCommand{\IS}{}{{IS}}
\RenewDocumentCommand{\IG}{}{{IG}}
\RenewDocumentCommand{\IGM}{}{{IG}}
\cs_new:Nn \@@_fpure_one:n
{
  f\c_math_subscript_token{#1}
  \peek_catcode_remove:NF \c_math_superscript_token
  { \c_math_superscript_token\bullet }
}
\RenewDocumentCommand{\fpure}{}
{
  \peek_catcode_remove:NTF \c_math_subscript_token
  { \@@_fpure_one:n }
  { f }
}
\cs_new:Nn \@@_intensive_two:n
{
  \c_math_subscript_token{#1}
  \peek_catcode:NF \c_math_superscript_token
  {
    \c_math_superscript_token\bullet
  }
}
\cs_set:Nn \@@_intensive:n
{ #1
  \peek_catcode_remove:NT \c_math_subscript_token
  { \@@_intensive_two:n }
}
\cs_new:Nn \@@_phipure_one:n
{
  \phi\c_math_subscript_token{#1}
  \peek_catcode:NF \c_math_superscript_token
  { \c_math_superscript_token\bullet }
}
\RenewDocumentCommand{\phipure}{}
{
  \peek_catcode_remove:NTF \c_math_subscript_token
  {
    \@@_phipure_one:n
  }
  {\phi}
}
\RenewDocumentCommand{\mixing}{}{{MIX}}
\RenewDocumentCommand{\Deltamix}{m}
    {\Delta\c_math_subscript_token\mixing #1}
\RenewDocumentCommand{\Deltarxn}{m}{\Delta #1}
\RenewDocumentCommand{\allbut}{O{j} m m}
{ \tl_if_eq:nnTF {#1} {#2}
  {
    {#3}\c_math_subscript_token k\neq{#3}\c_math_subscript_token{#2}
  }
```

```
      {
        {#3}\c_math_subscript_token{#1}\neq{#3}\c_math_subscript_token{#2}
      }
    }
  }
}
```

We execute the default options below.

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

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

```
\tl_const:Nn \c_@@_sort_order_tl
    {\Et\Em\Es\Ut\Um\Us\Ht\Hm\Hs\Ft\Fm\Fs\Gt\Gm\Gs\Lt\Lm\Ls T\St\Sm\Ss
        P\Vt\Vm\Vs\mu\Nt mwxyz\At\Am\As\sigma
        ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklnopqrstuvwxyz}
\clist_new:N \l_@@_in_list_clist
\clist_new:N \l_@@_sorted_list_clist
\clist_new:N \l_@@_remaining_list_clist
\cs_new:Nn \@@_sort_clist:n
{% Sort the list in the order of \c_@@_sort_order_tl

  % Wipe out any remnants from the last sort
  \clist_clear:N \l_@@_in_list_clist
  \clist_clear:N \l_@@_sorted_list_clist

  % Make a copy of the list
  \clist_set:Nn \l_@@_remaining_list_clist {#1}

  % Make a list of everything that's in the known sort order list
  % and put everything else in the "not in sort order list" list.
  \tl_map_inline:Nn \c_@@_sort_order_tl
  {
    \clist_if_in:NnT \l_@@_remaining_list_clist {##1}
    { \clist_put_right:Nn \l_@@_in_list_clist {##1} }

    \clist_remove_all:Nn \l_@@_remaining_list_clist {##1}
  }

  % Then merge the lists back together again.
  \clist_if_empty:NF \l_@@_in_list_clist
  {
    \clist_put_right:Nn \l_@@_sorted_list_clist \l_@@_in_list_clist
  }
  \clist_if_empty:NF \l_@@_remaining_list_clist
  {
    \clist_put_right:Nn \l_@@_sorted_list_clist \l_@@_remaining_list_clist
  }
  \clist_use:Nn \l_@@_sorted_list_clist ,
}
```

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.

```
\dim_new:N \l_@@_Partial_const_dim
\dim_new:N \l_@@_operator_width_dim
\dim_new:N \l_@@_adjust_width_dim
\settowidth{\l_@@_operator_width_dim}{=}
\dim_set:Nn \l_@@_adjust_width_dim {0.1\l_@@_operator_width_dim}
\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.

```
\tl_new:N \l_@@_Partial_start_tl
\tl_new:N \l_@@_Partial_end_tl
\tl_new:N \l_@@_Partial_empty_end_tl
\tl_new:N \l_@@_Partial_middle_tl
\tl_set:Nn \l_@@_Partial_start_tl {\left\l_@@_PartialOpen_tl}
\tl_set:Nn \l_@@_Partial_end_tl {\right\l_@@_PartialClose_tl}
\tl_set:Nn \l_@@_Partial_empty_end_tl {\right\l_@@_PartialEmptyClose_tl}
\tl_set:Nn \l_@@_Partial_middle_tl {\middle}
\cs_set_eq:NN \@@_frac:nn \frac
\NewDocumentCommand{\Partial}{s m m m}
{ \bool_if:nTF {#1}
  {% Starred form (recursive)
    \settowidth{\l_@@_Partial_const_dim}{\ensuremath{#4}}%
    \dim_add:Nn \l_@@_Partial_const_dim {-0.20\l_@@_Partial_const_dim}%
    \Partial{#2}{#3}{#4}%
    \bool_if:NT \l_@@_subscripted_bool
    { \dim_compare:nNnTF \l_@@_operator_width_dim
                         < \l_@@_Partial_const_dim
      { \kern -\l_@@_operator_width_dim }
      { \kern -\l_@@_Partial_const_dim }
    }
  }
  {% Unstarred form
    \bool_if:NTF \l_@@_subscripted_bool
    {% Handle case of empty variables held constant
      \tl_if_eq:nnTF {#4} {}
      { \l_@@_Partial_start_tl
        \@@_frac:nn{\partial #2}{\partial #3}\l_@@_Partial_empty_end_tl
      }
      { \l_@@_Partial_start_tl\@@_frac:nn{\partial #2}
                                         {\partial #3}\l_@@_Partial_end_tl
        \c_math_subscript_token{#4}%
      }
    }
    {% Check whether #4 contains \allNsbut{i} and #3 is \Nt_i
      \tl_if_in:nnTF {#3} {\Nt}
      { \RenewDocumentCommand{\allbut}{O{j} m m}{\allcomponents{##3}}
        \l_@@_Partial_start_tl
```

```
        \@@_frac:nn{\partial #2(\@@_sort_clist:n{#4})}
                   {\partial #3}\l_@@_Partial_end_tl
      }
      { \l_@@_Partial_start_tl
        \@@_frac:nn{\partial #2(\@@_sort_clist:n{#3,#4})}
                   {\partial #3}\l_@@_Partial_end_tl
      }
    }
  }
}
```

\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.

```
\NewDocumentCommand{\PartialBigg}{}
  { \tl_set:Nn \l_@@_Partial_start_tl {\Biggl\l_@@_PartialOpen_tl}
    \tl_set:Nn \l_@@_Partial_end_tl {\Biggr\l_@@_PartialClose_tl}
    \tl_set:Nn \l_@@_Partial_Empty_end_tl
        {\Biggr\l_@@_PartialEmptyClose_tl}
    \Partial
  }
```

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

```
\NewDocumentCommand{\Partialbigg}{}
  { \tl_set:Nn \l_@@_Partial_start_tl {\biggl\l_@@_PartialOpen_tl}
    \tl_set:Nn \l_@@_Partial_end_tl {\biggr\l_@@_PartialClose_tl}
    \tl_set:Nn \l_@@_Partial_empty_end_tl
        {\biggr\l_@@_PartialEmptyClose_tl}
    \Partial
  }
```

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

```
\NewDocumentCommand{\PartialSecond}{s m m m}
{
  \bool_if:nTF {#1}
  {% Starred form
    \settowidth{\l_@@_Partial_const_dim}{\ensuremath{#4}}%
    \dim_add:Nn \l_@@_Partial_const_dim {-0.20\l_@@_Partial_const_dim}
    \PartialSecond{#2}{#3}{#4}%
    \bool_if:nT \l_@@_subscripted_bool
    {  \dim_compare:nNnTF {\l_@@_operator_width_dim}
                          < {\l_@@_Partial_const_dim}
      { \kern -\l_@@_operator_width_dim }
      { \kern -\l_@@_Partial_const_dim }
    }
  }
  {% Unstarred form
    \bool_if:NTF \l_@@_subscripted_bool
    {% Handles case of empty variables held constant
      \tl_if_eq:nnTF {#4} {}
      { \l_@@_Partial_start_tl
            \@@_frac:nn{\partial\c_math_superscript_token 2 #2}
        {\partial #3\c_math_superscript_token 2}\l_@@_Partial_empty_end_tl
      }
      { \l_@@_Partial_start_tl
```

```
        \@@_frac:nn{\partial\c_math_superscript_token 2 #2}
            {\partial #3\c_math_superscript_token 2}\l_@@_Partial_end_tl
            \c_math_subscript_token{#4}%
      }
    }
    {% Check whether #4 contains \allNsbut{i} and #3 is \Nt_i
      \tl_if_in:nnTF {#2} {\Nt}
      { \RenewDocumentCommand{\allbut}{O{j} m m}{\allcomponents{##3}}
        \l_@@_Partial_start_tl
        \@@_frac:nn{\partial\c_math_superscript_token 2 #2(#4)}
            {\partial\c_math_superscript_token 2 #3}\l_@@_Partial_end_tl
      }
      { \l_@@_Partial_start_tl
        \@@_frac:nn{\partial\c_math_superscript_token 2
                    #2(\@@_sort_clist:n{#3,#4})}
            {\partial #3\c_math_superscript_token 2}\l_@@_Partial_end_tl
      }
    }
  }
}
```

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

```
\NewDocumentCommand{\PartialSecondBigg}{}
{ \tl_set:Nn \l_@@_Partial_start_tl {\biggl\l_@@_PartialOpen_tl}
  \tl_set:Nn \l_@@_Partial_end_tl {\biggl\l_@@_PartialClose_tl}
  \tl_set:Nn \l_@@_Partial_empty_end_tl {\biggl\l_@@_PartialClose_tl}
  \PartialSecond
}
```

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

```
\NewDocumentCommand{\PartialSecondbigg}{}
{ \tl_set:Nn \l_@@_Partial_start_tl {\biggl\l_@@_PartialOpen_tl}
  \tl_set:Nn \l_@@_Partial_end_tl {\biggl\l_@@_PartialClose_tl}
  \tl_set:Nn \l_@@_Partial_empty_end_tl {\biggl\l_@@_PartialClose_tl}
  \PartialSecond
}
```

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

```
\bool_new:N \l_@@_has_x_or_y_bool
\NewDocumentCommand{\PartialMixSecond}{s m m m m}
{
  \bool_if:nTF {#1}
  {% Starred version
    \settowidth{\l_@@_Partial_const_dim}{\ensuremath{#4}}%
    \dim_add:Nn \l_@@_Partial_const_dim {-0.20\l_@@_Partial_const_dim}
    \PartialMixSecond{#2}{#3}{#4}{#5}
    \bool_if:nT \l_@@_subscripted_bool
    { \dim_compare:nNnTF {\l_@@_operator_width_dim}
                         < {\l_@@_Partial_const_dim}
      { \kern -\l_@@_operator_width_dim }
      { \kern -\l_@@_Partial_const_dim }
    }
  }
```

```
{% Unstarred version
  \bool_if:nTF \l_@@_subscripted_bool
  {% subscripted version
    \tl_if_eq:nnTF {#5} {}
    {% Handle case of empty variables held constant
      \l_@@_Partial_start_tl
      \@@_frac:nn{\partial\c_math_superscript_token 2 #2}
          {\partial #3\partial #4}\l_@@_Partial_empty_end_tl
    }
    { \l_@@_Partial_start_tl
      \@@_frac:nn{\partial\c_math_superscript_token 2 #2}
          {\partial #3\partial #4}\l_@@_Partial_end_tl
          \c_math_subscript_token{#5}
    }
  }
  {% not subscripted
    \tl_if_eq:nnTF {#5} {}
    {% empty argument
      \l_@@_Partial_start_tl
      \@@_frac:nn{\partial\c_math_superscript_token 2
                      #2(\@@_sort_clist:n{#3,#4,#5})}
                  {\partial #3\partial #4}\l_@@_Partial_empty_end_tl
    }
    {% Check whether #3 OR #4 are \Nt_i/etc.
      \tl_if_in:nnTF {#3} {\Nt}
      { \RenewDocumentCommand{\allbut}{O{j} m m}{\allcomponents{##3}}%
        \l_@@_Partial_start_tl
        \@@_frac:nn{\partial\c_math_superscript_token 2
                      #2(\@@_sort_clist:n{#4,#5})}
                    {\partial #3\partial #4}\l_@@_Partial_end_tl
      }
      { \tl_if_in:nnTF {#4} {\Nt}
        { \RenewDocumentCommand{\allbut}{O{j} m m}{\allcomponents{##3}}%
          \l_@@_Partial_start_tl
          \@@_frac:nn{\partial\c_math_superscript_token 2
                        #2(\@@_sort_clist:n{#3,#5})}
                      {\partial #3\partial #4}\l_@@_Partial_end_tl
        }
        {% Check for x, y, or w
          \bool_set_false:N \l_@@_has_x_or_y_bool
          \tl_if_in:nnT {#3} {x}
          { \l_@@_has_x_or_y_bool }
          \tl_if_in:nnT {#3} {y}
          { \l_@@_has_x_or_y_bool }
          \tl_if_in:nnT {#3} {w}
          { \l_@@_has_x_or_y_bool }
          \bool_if:NTF \l_@@_has_x_or_y_bool
          { \RenewDocumentCommand{\allbutlastand}{O{j} m m}
              {\allcomponents{##3}}
            \l_@@_Partial_start_tl
            \@@_frac:nn{\partial\c_math_superscript_token 2
                        #2(\@@_sort_clist:n{#4,#5})}
                      {\partial #3\partial #4}\l_@@_Partial_end_tl
          }
          {
            \l_@@_Partial_start_tl
```

```
              \@@_frac:nn{\partial\c_math_superscript_token 2
                            #2(\@@_sort_clist:n{#3,#4,#5})}
                          {\partial #3\partial #4}\l_@@_Partial_end_tl
            }
          }
        }
      }
    }
}
```

\PartialMixSecondBigg
\PartialMixSecondbigg

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

```
\NewDocumentCommand{\PartialMixSecondBigg}{}
{ \tl_set:Nn \l_@@_Partial_start_tl {\Biggl\l_@@_PartialOpen_tl}
  \tl_set:Nn \l_@@_Partial_end_tl {\Biggl\l_@@_PartialClose_tl}
  \tl_set:Nn \l_@@_Partial_empty_end_tl {\Biggl\l_@@_PartialClose_tl}
  \PartialMixSecond
}
\NewDocumentCommand{\PartialMixSecondbigg}{}
{ \tl_set:Nn \l_@@_Partial_start_tl {\biggl\l_@@_PartialOpen_tl}
  \tl_set:Nn \l_@@_Partial_end_tl {\biggl\l_@@_PartialClose_tl}
  \tl_set:Nn \l_@@_Partial_empty_end_tl {\biggl\l_@@_PartialClose_tl}
  \PartialMixSecond
}
```

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.

```
\AtBeginDocument{%
  \@ifpackageloaded{amsmath}{}{%
      \PackageWarningNoLine{thermodynamics}
        {Package~amsmath~not~loaded;~load~to~make~PartialBigg~and~friends
            ~work~correctly}%
      \cs_set_eq:NN \PartialBigg \Partial
      \cs_set_eq:NN \Partialbigg \Partial
      \cs_set_eq:NN \PartialSecondBigg \PartialSecond
      \cs_set_eq:NN \PartialSecondbigg \PartialSecond
      \cs_set_eq:NN \PartialMixSecondBigg \PartialMixSecond
      \cs_set_eq:NN \PartialMixSecondbigg \PartialMixSecond
      \ProvideDocumentCommand{\rvert}{}{|}
      \ProvideDocumentCommand{\lvert}{}{|}
  }%
}
```

\Partialinline
\PartialSecondinline
\PartialMixSecondinline

Inline partial derivatives.

```
\NewDocumentCommand{\Partialinline}{}
{
  \cs_set:Nn \@@_frac:nn { ##1 \l_@@_Partial_middle_tl / ##2 }
  \Partial
}
\NewDocumentCommand{\PartialSecondinline}{}
{
  \cs_set:Nn \@@_frac:nn { ##1 \l_@@_Partial_middle_tl / ##2 }
  \PartialSecond
```

```
}
\NewDocumentCommand{\PartialMixSecondinline}{}
{
  \cs_set:Nn \@@_frac:nn { ##1 \l_@@_Partial_middle_tl / ##2 }
  \PartialMixSecond
}
```

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

```
\NewDocumentCommand{\Partialinlinetext}{}
{ \cs_set_eq:NN \l_@@_Partial_start_tl \l_@@_PartialOpen_tl
  \cs_set_eq:NN \l_@@_Partial_end_tl \l_@@_PartialClose_tl
  \cs_set_eq:NN \l_@@_Partial_empty_end_tl \l_@@_PartialEmptyClose_tl
  \cs_set_eq:NN \l_@@_Partial_middle_tl \relax
  \Partialinline
}
\NewDocumentCommand{\PartialSecondinlinetext}{}
{ \cs_set_eq:NN \l_@@_Partial_start_tl \l_@@_PartialOpen_tl
  \cs_set_eq:NN \l_@@_Partial_end_tl \l_@@_PartialClose_tl
  \cs_set_eq:NN \l_@@_Partial_empty_end_tl \l_@@_PartialEmptyClose_tl
  \cs_set_eq:NN \l_@@_Partial_middle_tl \relax
  \PartialSecondinline
}
\NewDocumentCommand{\PartialMixSecondinlinetext}{}
{ \cs_set_eq:NN \l_@@_Partial_start_tl \l_@@_PartialOpen_tl
  \cs_set_eq:NN \l_@@_Partial_end_tl \l_@@_PartialClose_tl
  \cs_set_eq:NN \l_@@_Partial_empty_end_tl \l_@@_PartialEmptyClose_tl
  \cs_set_eq:NN \l_@@_Partial_middle_tl \relax
  \PartialMixSecondinline
}
```

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.

```
\NewDocumentEnvironment{thermoparentheses}{}
{ \cs_set:Nn \l_@@_PartialOpen_tl {(}
  \cs_set:Nn \l_@@_PartialClose_tl {)}
  \cs_set:Nn \l_@@_PartialEmptyClose_tl {)}
}{}
```

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

```
\NewDocumentEnvironment{thermobrackets}{}
{ \tl_set:Nn \l_@@_PartialOpen_tl {[}
  \tl_set:Nn \l_@@_PartialClose_tl {]}
  \tl_set:Nn \l_@@_PartialEmptyClose_tl {]}
}{}
```

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

```
\NewDocumentEnvironment{thermobraces}{}
{ \tl_set:Nn \l_@@_PartialOpen_tl {\{}
  \tl_set:Nn \l_@@_PartialClose_tl {\}}
  \tl_set:Nn \l_@@_PartialEmptyClose_tl {\}}
}{}
```

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

```
\NewDocumentEnvironment{thermobar}{}
{ \tl_set:Nn \l_@@_PartialOpen_tl {.}
  \tl_set:Nn \l_@@_PartialClose_tl {\rvert}
  \tl_set:Nn \l_@@_PartialEmptyClose_tl {.}
}{}
```

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

```
\NewDocumentEnvironment{thermoplain}{}
{ \tl_set:Nn \l_@@_PartialOpen_tl {.}
  \tl_set:Nn \l_@@_PartialClose_tl {.}
  \tl_set:Nn \l_@@_PartialEmptyClose_tl {.}
  \bool_set_false:N \l_@@_subscripted_bool
}{}
```

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

```
\NewDocumentEnvironment{thermoNOsubscripts}{}
    {\bool_set_false:N \l_@@_subscripted_bool}
    {}
```

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

```
\NewDocumentEnvironment{thermosubscripts}{}
    {\bool_set_true:N \l_@@_subscripted_bool}
    {}
```

`thermomolesrange` (*env.*) Inside this environment, the macro `\allNs` will expand to $n_1, \ldots, n_C$ (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.

```
\NewDocumentEnvironment{thermomolesrange}{}
    { \@@_set_moles_range }
    {}
```

`thermointensiveplain` (*env.*)

```
%    \begin{macrocode}
\NewDocumentEnvironment{thermointensiveplain}{}
    { \@@_set_intensive_plain }
    {}
```

`thermoextensiveplain` (*env.*)

```
\NewDocumentEnvironment{thermoextensiveplain}{}
    { \@@_set_extensive_plain }
    {}
```

thermointensivelowercase (*env.*)

```
\NewDocumentEnvironment{thermointensivelowercase}{}
    {% {
% \RenewExpandableDocumentCommand{\MacroFont}{}{
%   \fontencoding\encodingdefault
%   \fontfamily\ttdefault
%   \fontseries\mddefault
%   \fontshape\shapedefault
%   \footnotesize}

      \@@_set_intensive_lowercase
      \@@_set_lowercase_pms
    }
    {}
```

thermoextensivesuperscript (*env.*)

```
\NewDocumentEnvironment{thermoextensivesuperscript}{}
    {
      \@@_set_extensive_superscripts
    }
    {}
```

4.6 User-Interface Macros to Define Symbols

\NewSubscriptedSymbol First, we define a command that serves to create "subscripted" symbols; for example, typing \cP_i should yield $C_{P,i}$ rather than C_{Pi}, $C_{P\,i}$, or C_{P_i}. Superscripts are also handled properly and can be in either order.

```
\cs_new:Npn \@@_check_definable:nN #1#2
{
  \bool_set_true:N \l_@@_arg_legal_bool
  \tl_trim_spaces_apply:nN {#1} \tl_if_single_token:nTF
  {
    \str_set:Nx \l_tmp_str {\tl_to_str:n {#1}}
    \int_compare:nNnT {\str_count:N \l_tmp_str} = 1
    { \PackageError{thermodynamics}
        {First~argument~of~'\tl_trim_spaces:o {\tl_to_str:n {#2}}'~
         must~be~a~command}
        {The~first~argument~of~'\tl_trim_spaces:o {\tl_to_str:n {#2}}'~
         should~be~the~macro~that~will~be~used~to~refer~to~the~symbol.~
         The~provided~argument~'\tl_trim_spaces:o {\tl_to_str:n {#1}}'~
         is~a~single~character.
         \MessageBreak Perhaps~a~backslash~is~missing?}
    }
  }
  { \PackageError{thermodynamics}
      {First~argument~of~'\tl_trim_spaces:o {\tl_to_str:n {#2}}'~
       must~be~a~command}
      {The~first~argument~of~'\tl_trim_spaces:o {\tl_to_str:n {#2}}'~
       should~be~the~macro~that~will~be~used~to~refer~to~the~symbol.~
       The~provided~argument~'\tl_trim_spaces:o {\tl_to_str:n {#1}}'~
       contains~more~than~one~token.
       \MessageBreak Perhaps~a~backslash~is~missing?}
  }
}
\NewDocumentCommand{\NewSubscriptedSymbol}{m m m}
```

```
{
  \@@_check_definable:nN {#1} \NewSubscriptedSymbol
  \cs_if_exist:NT #1
  { \PackageError{thermodynamics}
      {Command~'\tl_trim_spaces:o {\tl_to_str:n {#1}}'~already~defined}
      {You~have~used~
       '\tl_trim_spaces:o {\tl_to_str:n {\NewSubscriptedSymbol}}'~
       with~a~command~that~already~has~a~definition}
  }
  \cs_new:cpn {\cs_to_str:N #1_one:n} ##1
  {
    {#2}\c_math_superscript_token{##1}
    \peek_catcode_remove:NTF \c_math_subscript_token
    { \use:c {\cs_to_str:N #1_three:n} }
    { \c_math_subscript_token{#3} }
  }

  \cs_new:cpn {\cs_to_str:N #1_two:n} ##1
  { {#2}\c_math_subscript_token{#3,##1} }

  \cs_new:cpn {\cs_to_str:N #1_three:n} ##1
  { \c_math_subscript_token{#3,##1} }

  \NewDocumentCommand{#1}{}
  {% @branch
    \peek_catcode_remove:NTF \c_math_superscript_token
    { \use:c {\cs_to_str:N #1_one:n} }
    { \peek_catcode_remove:NTF \c_math_subscript_token
      { \use:c {\cs_to_str:N #1_two:n} }
      { {#2}\c_math_subscript_token{#3} }
    }
  }
}
\NewDocumentCommand{\RenewSubscriptedSymbol}{m m m}
{
  \@@_check_definable:nN {#1} \RenewSubscriptedSymbol
  \cs_if_exist:NF #1
  { \PackageError{thermodynamics}
      {Command~'\tl_trim_spaces:o {\tl_to_str:n {#1}}'~not~defined}
      {You~have~used~
       '\tl_trim_spaces:o {\tl_to_str:n {\RenewSubscriptedSymbol}}'~
       with~a~command~that~does~not~have~a~definition}
  }
  \cs_set:cpn {\cs_to_str:N #1_one:n} ##1
  {
    {#2}\c_math_superscript_token{##1}
    \peek_catcode_remove:NTF \c_math_subscript_token
    { \use:c {\cs_to_str:N #1_three:n} }
    { \c_math_subscript_token{#3} }
  }

  \cs_set:cpn {\cs_to_str:N #1_two:n} ##1
  { {#2}\c_math_subscript_token{#3,##1} }

  \cs_set:cpn {\cs_to_str:N #1_three:n} ##1
  { \c_math_subscript_token{#3,##1} }
```

```
  \RenewDocumentCommand{#1}{}
  {% @branch
    \peek_catcode_remove:NTF \c_math_superscript_token
    { \use:c {\cs_to_str:N #1_one:n} }
    { \peek_catcode_remove:NTF \c_math_subscript_token
      { \use:c {\cs_to_str:N #1_two:n} }
      { {#2}\c_math_subscript_token{#3} }
    }
  }

}
```

\NewSuperscriptedSymbol

```
\NewDocumentCommand{\NewSuperscriptedSymbol}{m m m}
{
  \@@_check_definable:nN {#1} \NewSuperscriptedSymbol
  \cs_if_exist:NT #1
  { \PackageError{thermodynamics}
      {Command~'\tl_trim_spaces:o {\tl_to_str:n {#1}}'~already~defined}
      {You~have~used~
       '\tl_trim_spaces:o {\tl_to_str:n {\NewSuperscriptedSymbol}}'~
       with~a~command~that~already~has~a~definition}
  }
  \cs_new:cpn {\cs_to_str:N #1_one:n} ##1
  {
    {#2}\c_math_subscript_token{#1}
    \peek_catcode_remove:NTF \c_math_superscript_token
    { \use:c {\cs_to_str:N #1_one:n} }
    { \c_math_subscript_token{#3} }
  }

  \cs_new:cpn {\cs_to_str:N #1_two:n} ##1
  { {#2}\c_math_superscript_token{#3,##1} }

  \cs_new:cpn {\cs_to_str:N #1_three:n} ##1
  { {#2}\c_math_superscript_token{#3,##1} }

  \NewDocumentCommand{#1}{}
  {
    \peek_catcode_remove:NTF \c_math_subscript_token
    { \use:c {\cs_to_str:N #1_one:n} }
    { \peek_catcode_remove:NTF \c_math_superscript_token
      { \use:c {\cs_to_str:N #1_two:n} }
      { {#2}\c_math_superscript_token{#3} }
    }
  }
}
\NewDocumentCommand{\RenewSuperscriptedSymbol}{m m m}
{
  \@@_check_definable:nN {#1} \RenewSuperscriptedSymbol
  \cs_if_exist:NF #1
  { \PackageError{thermodynamics}
      {Command~'\tl_trim_spaces:o {\tl_to_str:n {#1}}'~not~defined}
      {You~have~used~
       '\tl_trim_spaces:o {\tl_to_str:n {\RenewSuperscriptedSymbol}}'~
```

```
      with~a~command~that~does~not~have~a~definition}
  }
  \cs_set:cpn {\cs_to_str:N #1_one:n} ##1
  {
    {#2}\c_math_subscript_token{##1}
    \peek_catcode_remove:NTF \c_math_superscript_token
    { \use:c {\cs_to_str:N #1_three:n} }
    { \c_math_superscript_token{#3} }
  }

  \cs_set:cpn {\cs_to_str:N #1_two:n} ##1
  { {#2}\c_math_superscript_token{#3,##1} }

  \cs_set:cpn {\cs_to_str:N #1_three:n} ##1
  { \c_math_superscript_token{#3,##1} }

  \RenewDocumentCommand{#1}{}
  {% @branch
    \peek_catcode_remove:NTF \c_math_subscript_token
    { \use:c {\cs_to_str:N #1_one:n} }
    { \peek_catcode_remove:NTF \c_math_superscript_token
      { \use:c {\cs_to_str:N #1_two:n} }
      { {#2}\c_math_superscript_token{#3} }
    }
  }
}
```

\heatcapacitysymbol Now we define symbols for the heat capacities, compressibilities, and so forth.
\compressibilitysymbol
\expansivitysymbol

```
\NewExpandableDocumentCommand{\heatcapacitysymbol}{}{C}
\NewExpandableDocumentCommand{\compressibilitysymbol}{}{\kappa}
\NewExpandableDocumentCommand{\expansivitysymbol}{}{\alpha}
\NewExpandableDocumentCommand{\JTsymbol}{}{\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": `\cV_i` will recognize the subscript appropriately and render $C_{V,i}$ rather than $C_{V}i$ or some other unintended symbol. Superscripts are also handled appropriately and can be in either order.

```
\NewSubscriptedSymbol{\cV}{\@@_intensive:n \heatcapacitysymbol}
    {\g_@@_volume_symbol}
\NewSubscriptedSymbol{\cP}{\@@_intensive:n \heatcapacitysymbol}
    {\g_@@_pressure_symbol}
```

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

```
\NewSubscriptedSymbol{\cVt}{\@@_extensive:n \heatcapacitysymbol}
    {\g_@@_volume_symbol}
\NewSubscriptedSymbol{\cPt}{\@@_extensive:n \heatcapacitysymbol}
    {\g_@@_pressure_symbol}
```

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

```
\NewSubscriptedSymbol{\cVs}{\@@_specific:n \heatcapacitysymbol}
    {\g_@@_volume_symbol}
\NewSubscriptedSymbol{\cPs}{\@@_specific:n \heatcapacitysymbol}
    {\g_@@_pressure_symbol}
```

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

```
\NewSubscriptedSymbol{\kappaT}{\compressibilitysymbol}
    {\g_@@_temperature_symbol}
\NewSubscriptedSymbol{\kappaS}{\compressibilitysymbol}
    {\g_@@_entropy_symbol}
```

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

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

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

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

\Psat \Pvap \phisat \fsat \sat The \Psat macro (and its clone, the \Pvap macro) should be used for the saturation pressure. Similarly, a \phisat macro typesets the fugacity coefficient at saturation. The \fsat macro similarly renders the fugacity at saturation. Other saturation properties should use M^\sat or similar, preferably by defining another macro such as \Gmsat.

```
\NewDocumentCommand{\sat}{}{{\text{sat}}}
\NewDocumentCommand{\Psat}{}{P\c_math_superscript_token\sat}
\NewDocumentCommand{\Pvap}{}{\Psat}
\NewDocumentCommand{\phisat}{}{\phi\c_math_superscript_token\sat}
\NewDocumentCommand{\fsat}{}{\fpure\c_math_superscript_token\sat}
```

\std \Pstd \fstd The \std macro denotes standard properties. \Pstd and \fstd are defined for convenience and for consistency across textbooks.

```
\NewDocumentCommand{\std}{}{\circ}
\NewDocumentCommand{\Pstd}{}{P\c_math_superscript_token \std}
\NewDocumentCommand{\fstd}{}{f\c_math_superscript_token \std}
```

\Deltamix \mixing Mixing properties, such as ΔG_{mix}, should be accessed using \Deltamix\Gm and similar constructions—this construct will typeset as $\Delta_{MIX}G$ using the Thompson package option, for example, and as $\Delta_{\text{mix}}\underline{G}$ using the Sandler package option.

```
\NewDocumentCommand{\mixing}{}{{\text{mix}}}
\NewDocumentCommand{\Deltamix}{m}
    {\Delta{#1}\c_math_subscript_token\mixing}
```

\Deltafus \fusion \Deltavap \vaporization \Deltasub \sublimation \Deltarxn \reaction Similar entities for property changes on fusion, reaction, sublimation, and vaporization are defined.

```
\NewDocumentCommand{\fusion}{}{{\text{fus}}}
\NewDocumentCommand{\reaction}{}{{\text{rxn}}}
\NewDocumentCommand{\sublimation}{}{{\text{sub}}}
\NewDocumentCommand{\vaporization}{}{{\text{vap}}}
\NewDocumentCommand{\formation}{}{f}
\NewDocumentCommand{\Deltafus}{m}
    {\Delta #1\c_math_superscript_token\fusion}
\NewDocumentCommand{\Deltasub}{m}
    {\Delta #1\c_math_superscript_token\sublimation}
\NewDocumentCommand{\Deltavap}{m}
    {\Delta #1\c_math_superscript_token\vaporization}
\NewDocumentCommand{\Deltarxn}{m}
```

```
    {\Delta #1\c_math_subscript_token\reaction}
\tl_new:N \l_@@_Deltaf_sym_tl
\NewSubscriptedSymbol{\@@_Deltaf}{\l_@@_Deltaf_sym_tl}{f}
\NewDocumentCommand{\Deltaf}{m}
{ \tl_set:Nn \l_@@_Deltaf_sym_tl {\Delta #1}
  \@@_Deltaf
}
```

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

```
\NewDocumentCommand{\fmix}{}{\hat{f}}
```

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

```
\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, `\phipure`, for fugacity coefficients.

```
\NewDocumentCommand{\fpure}{}{f}
\NewDocumentCommand{\phipure}{}{\phi}
```

4.7 Partial Molar Quantities

`\partialmolar` Partial molar quantities with superscripts appear as $\overline{G_i^{\text{IG}}}$ or $\overline{G_i^R}$, rather than something like $\overline{G}_i^{\text{IG}}$ or $\overline{G}_i^R$; 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^{\IGM}_i`, `\Gpm^{\IGM}_i`, and even `\Gpm_i^{\IGM}`; they can also be treated as commands: `\Gpm{i}` is equivalent to `\Gpm_i` and `\Gpm[\IGM]{i}` is equivalent to `\Gpm_i^{\IGM}`. The macro `\partialmolar` can be used to create an arbitrary partial molar symbol.

```
\tl_new:N \l_@@_pm_symbol_tl
\tl_new:N \l_@@_pm_arg_tl
\NewDocumentCommand{\partialmolar}{m}
{
  \tl_set:Nn \l_@@_pm_symbol_tl {#1}
  \@@_generic_pm:
}
%% cases to consider:
%% (1) \Mpm{i}
%% (2) \Mpm[S]{i}
%% (3) \Mpm^S_i
%% (4) \Mpm_i^S
%% (5) \Mpm_i
%% note that \Mpm^S with no subscript makes no sense and is thus forbidden
\cs_new:Nn \@@_generic_pm:
{
  \peek_catcode_remove:NTF \c_math_subscript_token
  {% case 4 or case 5
```

```
    \@@_pm_case_four_or_five
  }
  {% look for superscript token
    \peek_catcode_remove:NTF \c_math_superscript_token
    {% case 3: \Mpm^{#1}_{#2} or \Mpm^{#1}{#2}
      \@@_pm_case_three
    }
    {% Look for optional argument [...]
      \peek_charcode:NTF [
      {% case 2: \Mpm[S]{i}
        \@@_pm_case_two
      }
      {% case 1: \Mpm{i}
        \@@_pm_case_one
      }
    }
  }
}
\cs_new:Npn \@@_pm_case_one #1
{
  \@@_overline:n {\l_@@_pm_symbol_tl\c_math_subscript_token{#1}}
}
\cs_new:Npn \@@_pm_case_two [#1]#2
{
  \@@_overline:n {\l_@@_pm_symbol_tl
    \c_math_superscript_token{#1}\c_math_subscript_token{#2}}
}
\cs_new:Npn \@@_pm_case_three #1
{
  \tl_set:Nn \l_@@_pm_arg_tl {#1}
  \peek_catcode_remove:NTF \c_math_subscript_token
  { \@@_pm_case_three_part_two }
  { \@@_pm_case_three_part_two }
}
\cs_new:Npn \@@_pm_case_three_part_two #1
{
  \@@_overline:n {\l_@@_pm_symbol_tl
    \c_math_superscript_token{\l_@@_pm_arg_tl}
    \c_math_subscript_token{#1}}
}
\cs_new:Npn \@@_pm_case_four_or_five #1
{
  \tl_set:Nn \l_@@_pm_arg_tl {#1}
  \peek_catcode_remove:NTF \c_math_superscript_token
  { \@@_pm_case_four }
  { \@@_pm_case_five }
}
\cs_new:Npn \@@_pm_case_four #1
{
  \@@_overline:n {\l_@@_pm_symbol_tl\c_math_superscript_token{#1}
    \c_math_subscript_token{\l_@@_pm_arg_tl}}
}
\cs_new:Npn \@@_pm_case_five
{
  \@@_overline:n {\l_@@_pm_symbol_tl
    \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 $\underline{\mathcal{Q}}$, $\mathcal{Q}$, and $\hat{\mathcal{Q}}$, 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 $\overline{\Psi}_i$.

`\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 $\overline{S_i}$ (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.

```
\NewDocumentCommand{\NewThermodynamicProperty}{m m}
{
  \NewExtensiveProperty{#1}{#2}
  \NewPartialMolarProperty{#1}{#2}
  \NewResidualProperty{#1}{#2}
  \NewExcessProperty{#1}{#2}
}
\NewDocumentCommand{\NewExtensiveProperty}{m m}
{
  % Extensive property
  \exp_after:wN \NewDocumentCommand \exp_after:wN
    {\cs:w #1t\cs_end:}{}{\@@_extensive:n {#2}}
  % Molar property
  \exp_after:wN \NewDocumentCommand \exp_after:wN
    {\cs:w #1m\cs_end:}{}{\@@_intensive:n {#2}}
  % Specific property
  \exp_after:wN \NewDocumentCommand \exp_after:wN
    {\cs:w #1s\cs_end:}{}{\@@_specific:n {#2}}
}
```

```
\NewDocumentCommand{\NewPartialMolarProperty}{m m}
{
  % Partial molar property
  \exp_after:wN \NewDocumentCommand \exp_after:wN
    {\cs:w #1pm\cs_end:}{}{\partialmolar{#2}}
}
\NewDocumentCommand{\NewExcessProperty}{m m}
{
  \exp_after:wN \NewSuperscriptedSymbol \exp_after:wN
    {\cs:w #1 Et\cs_end:}{\@@_extensive:n{#2}}{\excess}
  \exp_after:wN \NewSuperscriptedSymbol \exp_after:wN
    {\cs:w #1 E\cs_end:}{\@@_intensive:n{#2}}{\excess}
  \exp_after:wN \NewSuperscriptedSymbol \exp_after:wN
    {\cs:w #1 Es\cs_end:}{\@@_specific:n{#2}}{\excess}

  %^^A TODO: make super- or subscripted partial molar quantities work
  % Excess partial molar property
  \exp_after:wN \NewDocumentCommand \exp_after:wN
    {\cs:w #1Epm\cs_end:}{}{\partialmolar{#2}
       \c_math_superscript_token\excess}
}
\NewDocumentCommand{\NewResidualProperty}{m m}
{
  \exp_after:wN \NewSuperscriptedSymbol \exp_after:wN
    {\cs:w #1 Rt\cs_end:}{\@@_extensive:n{#2}}{\residual}
  \exp_after:wN \NewSuperscriptedSymbol \exp_after:wN
    {\cs:w #1 R\cs_end:}{\@@_intensive:n{#2}}{\residual}
  \exp_after:wN \NewSuperscriptedSymbol \exp_after:wN
    {\cs:w #1 Rs\cs_end:}{\@@_specific:n{#2}}{\residual}

  % Residual partial molar property
  \exp_after:wN \NewDocumentCommand \exp_after:wN
    {\cs:w #1Rpm\cs_end:}{}{\partialmolar{#2}
        \c_math_superscript_token\residual}
}
```

\Nt
\Et
\Em
\Es
\Epm

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

```
\NewDocumentCommand{\Nt}{}{\g_@@_mole_symbol}
\NewThermodynamicProperty{E}{\g_@@_total_energy_symbol}
\NewThermodynamicProperty{U}{\g_@@_internal_energy_symbol}
\NewThermodynamicProperty{F}{\g_@@_Helmholtz_symbol}
\NewThermodynamicProperty{G}{\g_@@_Gibbs_symbol}
\NewThermodynamicProperty{H}{\g_@@_enthalpy_symbol}
\NewThermodynamicProperty{L}{\g_@@_Landau_symbol}
\NewThermodynamicProperty{V}{\g_@@_volume_symbol}
\NewThermodynamicProperty{S}{\g_@@_entropy_symbol}
\NewExtensiveProperty{A}{\g_@@_area_symbol}
\NewPartialMolarProperty{A}{\g_@@_area_symbol}
\NewExtensiveProperty{Q}{\g_@@_heat_symbol}
\NewExtensiveProperty{W}{\g_@@_work_symbol}
```

\cVpm
\cPpm

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

```
\NewDocumentCommand{\cPpm}{}
```

```
{
  \cs_set:Npn \@@_pm_case_one ##1
  {
    \@@_overline:n {\l_@@_pm_symbol_tl\c_math_subscript_token
        {\g_@@_pressure_symbol,##1}}
  }
  \cs_set:Npn \@@_pm_case_two [##1]##2
  {
    \@@_overline:n {\l_@@_pm_symbol_tl\c_math_superscript_token{##1}
        \c_math_subscript_token{\g_@@_pressure_symbol,##2}}
  }
  \cs_set:Npn \@@_pm_case_three_part_two ##1
  {
    \@@_overline:n {\l_@@_pm_symbol_tl
        \c_math_superscript_token{\l_@@_pm_arg_tl}
        \c_math_subscript_token{\g_@@_pressure_symbol,##1}}
  }
  \cs_set:Npn \@@_pm_case_four ##1
  {
    \@@_overline:n {\l_@@_pm_symbol_tl
        \c_math_superscript_token{##1}\c_math_subscript_token
      {\g_@@_pressure_symbol,\l_@@_pm_arg_tl}}
  }
  \cs_set:Npn \@@_pm_case_five
  {
    \@@_overline:n {\l_@@_pm_symbol_tl\c_math_subscript_token
        {\g_@@_pressure_symbol,\l_@@_pm_arg_tl}}
  }
  \partialmolar{\heatcapacitysymbol}
}
\NewDocumentCommand{\cVpm}{}
{
  \cs_set:Npn \@@_pm_case_one ##1
  {
    \@@_overline:n {\l_@@_pm_symbol_tl\c_math_subscript_token
        {\g_@@_volume_symbol,##1}}
  }
  \cs_set:Npn \@@_pm_case_two [##1]##2
  {
    \@@_overline:n {\l_@@_pm_symbol_tl
            \c_math_superscript_token{##1}\c_math_subscript_token
        {\g_@@_volume_symbol,##2}}
  }
  \cs_set:Npn \@@_pm_case_three_part_two ##1
  {
    \@@_overline:n {\l_@@_pm_symbol_tl
        \c_math_superscript_token{\l_@@_pm_arg_tl}
        \c_math_subscript_token{\g_@@_volume_symbol,##1}}
  }
  \cs_set:Npn \@@_pm_case_four ##1
  {
    \@@_overline:n {\l_@@_pm_symbol_tl
        \c_math_superscript_token{##1}
        \c_math_subscript_token
      {\g_@@_volume_symbol,\l_@@_pm_arg_tl}}
  }
```

```
  \cs_set:Npn \@@_pm_case_five
  {
    \@@_overline:n {\l_@@_pm_symbol_tl\c_math_subscript_token
        {\g_@@_volume_symbol,\l_@@_pm_arg_tl}}
  }
  \partialmolar{\heatcapacitysymbol}
}
```

4.9 Residual and Excess Properties

\residual \excess

Macros are defined for residual properties (departure from ideal gases) and 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.

```
\NewDocumentCommand{\residual}{}{R}
\NewDocumentCommand{\excess}{}{E}
```

\prodall \sumall \summallbutlast

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

```
\[ \sumall_i x_i = 1 \qquad \sumallbutlast_i x_i = 1 - x_\ncomponents
    \qquad K = \exp\left(\frac{-\Deltarxn\Gm^\std}{RT}\right)
             = \prodall_i a_i^{\nu_i} \]
```

gives

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

```
\NewDocumentCommand{\sumall}{m m}
  { \sum\c_math_subscript_token{#2=1}
      \c_math_superscript_token{\ncomponents} }
\NewDocumentCommand{\sumallbutlast}{m m}
  { \sum\c_math_subscript_token{#2=1}
      \c_math_superscript_token{\ncomponents-1} }
\NewDocumentCommand{\prodall}{m m}
  { \prod\c_math_subscript_token{#2=1}
       \c_math_superscript_token{\ncomponents} }
```

\IG \IGM \IS

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

```
\NewDocumentCommand{\IG}{}{{\text{IG}}}
\NewDocumentCommand{\IGM}{}{{\text{IGM}}}
\NewDocumentCommand{\IS}{}{{\text{IS}}}
```

\Henryrat \Henrymol

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

```
\NewDocumentCommand{\Henryrat}{}{h}
\NewDocumentCommand{\Henrymol}{}{\mathcal{H}}
```

\gammarat \gammamol

The ordinary activity coefficient is universally denoted γ, so I have not defined a 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 `\gammarat` as γ^* and `\gammamol` as $\gamma^\square$. `\gammamol` will use `\square` from packages if it is defined; if not, it "fakes it" with the definition below.

```
\AtBeginDocument{%
  \providecommand*{\square}{%
    \text{\leavevmode
      \hbox to.65em{%
      \hfil\vrule
      \vbox to.53em{\hrule width.45em\vfil\hrule}%
      \vrule\hfil}%
    }%
  }%
}
\NewDocumentCommand{\gammarat}{}{\gamma\c_math_superscript_token\ast}
\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).

```
\NewDocumentEnvironment{thermovmatrix}{}
{ \cs_if_exist:NTF \vmatrix
  { \begin{vmatrix} }
  { \left|\begin{array}{c c c c c c c c c c} }
}
{ \cs_if_exist:NTF \endvmatrix
  {  \end{vmatrix} }
  {  \end{array}\right| }
}
```

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

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

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

```
\NewDocumentCommand{\Jacobiandet}{O{} O{} m m}
{
  \@@_Jacobian_set_ncomponents:nn {#3} {#4}
  \begin{thermovmatrix}
    \@@_Jacobianmatrix:nnnn {#1} {#2} {#3} {#4}
  \end{thermovmatrix}
}
\seq_new:N \l_@@_row_seq
\seq_new:N \l_@@_matrix_seq
\clist_new:N \l_@@_other_vars_clist
\clist_new:N \l_@@_other_vars_copy_clist
\tl_new:N \l_@@_Jacobian_x_tl
\tl_new:N \l_@@_Jacobian_n_tl
\tl_new:N \l_@@_Jacobian_temp_tl
\bool_new:N \l_@@_found_dots_bool
\cs_new:Nn \@@_Jacobian_set_ncomponents:nn
{
  % If any entry is \dots, we assume the Jacobian is of the form
  % d(f_1,\dots,f_n)/d(x_1,\dots,x_n) where f is some function
  % (any symbol) and x is some variable (any symbol).
```

```
  \tl_if_in:nnTF {#1} {\dots}
  {% Has dots
    \bool_set_true:N \l_@@_found_dots_bool
    % look for what "x" is
    \tl_set:Nn \l_@@_Jacobian_x_tl {\tl_head:n {#2}}
    % look for what "n" is and set \ncomponents to it
    \tl_set:Nx \l_@@_Jacobian_n_tl {\tl_item:nn {#2} {-1}}
    \RenewExpandableDocumentCommand{\ncomponents}{}{\l_@@_Jacobian_n_tl}
  }
  {% Does not have dots; proceed accordingly
    \bool_set_false:N \l_@@_found_dots_bool
  }
}
\cs_new_protected:Nn \@@_Jacobianmatrix:nnnn
{
  \seq_clear:N \l_@@_matrix_seq
  \clist_set:Nn \l_@@_other_vars_clist {#4}
  \clist_set_eq:NN \l_@@_other_vars_copy_clist \l_@@_other_vars_clist

  \clist_map_inline:nn {#3}
  {
    \seq_clear:N \l_@@_row_seq
    \tl_if_in:nnTF {##1} {\dots}
    {% The current row has "dots" => row is \vdots && \vdots
      \seq_put_right:Nn \l_@@_matrix_seq
        { \vdots \c_alignment_token \c_alignment_token \vdots }
    }
    {% Ordinary row
      \clist_map_inline:nn {#4}
      {
        \tl_if_in:nnTF {####1} {\dots}
        {% this column has "dots" in it
          \seq_put_right:Nn \l_@@_row_seq \dots
        }
        {% Normal column
          \clist_set_eq:NN \l_@@_other_vars_clist
                           \l_@@_other_vars_copy_clist
          \clist_remove_all:Nn \l_@@_other_vars_clist {####1}
          \bool_if:NTF \l_@@_found_dots_bool
          { \tl_set:Nn \l_@@_Jacobian_temp_tl {\tl_item:nn {####1} {-1}}
            \seq_put_right:Nx \l_@@_row_seq
            {
              #1\Partial{##1}{####1}
                  {\allbut{\l_@@_Jacobian_temp_tl}{\l_@@_Jacobian_x_tl}}
            }
          }
          { \seq_put_right:Nx \l_@@_row_seq
            {
              #1\Partial{##1}{####1}
                        {\clist_use:Nn \l_@@_other_vars_clist ,}
            }
          }
        }
      }
      \seq_put_right:Nx \l_@@_matrix_seq
      {
```

```
        \seq_use:Nn \l_@@_row_seq { \c_alignment_token }
      }
    }
  }
  \tl_if_empty:nTF {#2}
  {
    \tl_if_eq:nnTF {#1} {\displaystyle}
    { \seq_use:Nn \l_@@_matrix_seq { \\[2.75ex] } }
    { \seq_use:Nn \l_@@_matrix_seq { \\[1.25ex] } }
  }
  {
    \seq_use:Nn \l_@@_matrix_seq { \\[#2] }
  }
}
```

Change History

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.

T

U

V

W