First order derivatives of thermodynamic functions under assumption of no chemical changes revisited

Lorentz Jäntschi [a,b,*], Sorana D. Bolboacă [c,d]

[a] Technical University of Cluj-Napoca, Department of Physics and Chemistry, Cluj-Napoca, Romania
[b] Babeş-Bolyai University, Institute for Doctoral Studies, Cluj-Napoca, Romania
[c] Iuliu Haţieganu University of Medicine and Pharmacy, Department of Medical Informatics and Biostatistics, Cluj-Napoca, Romania
[d] University of Agricultural Science and Veterinary Medicine Cluj-Napoca, Cluj-Napoca, Romania

A B S T R A C T

In the field of experimental chemistry, it is extremely useful and necessary to have fast access to equations that link the physical and chemical parameters to the state. In this regard, starting with Bridgmann's equations, an online application had been projected and implemented in order to generate all equations of first order derivatives of thermodynamic parameters for a closed system. MathJax open source and PHP language have been used for implementation. So a framework was conceived and implemented to automatically generate the first order derivatives of the thermodynamic parameters for closed system.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

The principles of thermodynamics as they are stated today combine classical thermodynamics whose formulation is based on experimental observation of macroscopic quantities measured in laboratory thermodynamic systems. To mark the history of classical thermodynamics, we have to go back to 1960, when Otto von Guericke created the first vacuum pump [1]. Thermodynamics has evolved as a science of gaseous states by establishing the (ideal) gas laws till apparition in 1761 of 'heat capacity' and 'latent heat' concepts [2]. At this point in time thermodynamics became the science of states and transformation of matter in any state of aggregation. More than 50 years later, Sadi Carnot published his reflections on the combustion engine [3]. Carnot's idealization (Carnot cycle) marked the apparition of a new approach: statistical thermodynamics. Successively, Clausius [4], Maxwell [5], Boltzmann [6], and Planck [7] defined the elements of statistical thermodynamics. Through a series of papers published between 1873 and 1876, Gibbs completes thermodynamics by adding its appropriate mathematical apparatus [8]. In 1914, Bridgman [9] showed that all relations between variations of thermodynamic quantities can be expressed as functions of three parameters: C_P – heat capacity (at constant pressure), α_V – coefficient of thermal expansion, and β_T – isothermal compressibility.

Based on the relations given by Bridgman, a program able to generate the variation equations between all fundamental thermodynamic quantities has been implemented. Our aim was to have all equations in the same web-page without needing any program or any 'add-on' in this sense.

2. Materials and methods

2.1. Thermodynamic parameters

The total differential of a function f of several variables is the sum of all partial differentials [10]. In the absence of chemical changes, all state thermodynamic parameters are total differentials and depend on two variables (Table 1).

Identifying the heat from reversible process $dQ_{rev} = T \cdot dS$ and the mechanical work derived from quasi-static processes $dw_{cvs} = -p \cdot dV$, also heat (dQ) and work (dw) thermodynamic quantities became total differentials. The condensed collection of the Bridgman's thermodynamic equations [9] was implemented in our web-page using a series of modification schemas to translate these equations in their first order partial derivatives [11] (Table 2).

For example, using isothermal compressibility (β_T) and isobaric dilatation (α_V) the following expressions could be identified: $VpT = -V \cdot \beta_T$, $VTp = V \cdot \alpha_V$, and rest of the combinations containing V (volume), p (pressure), and T (temperature) letters could result

Table 1
Differentials of thermodynamic parameters.

$dE = T \cdot dS - p \cdot dV + \Sigma_i \cdot \mu_i \cdot dn_i$	In the presence of chemical changes
$dn_i = 0$ for each i	Provided that no chemical changes
$dE = T \cdot dS - p \cdot dV$	In the presence of chemical changes

E = internal energy; S = entropy; V = volume; n = number of particles; p = pressure.

from these two expressions. If the function f is total energy (E), then using specific heat at constant volume ($C_V = ETV$) and internal pressure ($\pi_T = EVT$) and starting from equation numbered as 1 in Table 2 to equation numbered as 6, all remaining partial derivatives of the energy when any two of the temperatures (T), pressures (p) and volumes (V) are kept constant could be identified.

2.2. System implementation

The MathJax open source [12], JavaScript platform for displaying of mathematics, has been used to implement the Bridgman's differentials in a web-page. A program has been developed and implemented using PHP language [13] to simplify and automate displaying of the equations.

The web-page was structured in two main parts: introduction and main part of the page.

- Introductory elements have been included in the first part of the page to allow utilization of equations without navigation on specialized pages. This part of the web-page provides the measures and relations used in thermodynamics.
- The main part of the page contains the relationships between first order derivatives of thermodynamic quantities. As implementation it was chosen to create an 'array' with equations, in a form similar to that of Bridgman. The first two fields of the array contain the variable and constant quantities from the expression of differential (these two quantities are interchangeable [9]) while the third field of the array stored the compacted form (removing signs from the definition of differentials) of the differentials (Implementation 1).

Implementation 1. First part from the definition of the array containing half-relationships between thermodynamic quantities.

Step 0. Instantiate 'Bridgman' as an array (of arrays, see below)
```
bridgman=array();
```
Step 1. Add to 'Bridgman' array the list of half partial derivatives (see [9])
```
bridgman[]=array("T","p","1");
bridgman[]=array("V","p","VTp");
bridgman[]=array("S","p","HTp/T");
...//(45 equations in total)
```
To speed up the process of displaying the equations, interchanged equations can also be stored in this array (Implementation 2).

Implementation 2. First part from second half of the definition of the array containing half-relationships between thermodynamic quantities.

Step 2. Add to 'Bridgman' array the list of half partial derivatives with the reversed order of the variable and the constant ($\partial x|_{y=ct} = -\partial y|_{x=ct}$, see [9])
```
bridgman[]=array("p","T","-1");
bridgman[]=array("p","V","-VTp");
bridgman[]=array("p","S","-HTp/T");
...//(again 45 equations in total)
```

From this point, it is only a matter of manipulation of the equations, to provide in full the desired equations. Three algorithms have been implemented to accomplish this task (Algorithms 1–3).

Algorithm 1. Function used to generate a partial derivative.

Returns the code sequence (text-based) needed for MathJax to display a partial derivatives; take as input an array and returns a string containing the elements of the array according to the MathJax syntax.

```
function pd(x){
return("\left. \\frac{\partial"+x[0]+"}{\partial"+x[1]+"}
\\right | _{"+x[2]+"=ct.}");
}
```

An instance of the Algorithm 1 will produce first (left) part of an equation. For instance *pd(array('E','G','A'))* returns the text corresponding to the image of the derivative of the E by G when A is constant:

```
"\left. \\frac{\partial E}{\partial G} \\right | _{A=ct.}"
```
text

$\left. \dfrac{\partial E}{\partial G} \right|_{A=ct.}$
image

Algorithm 2. This algorithm implements the first function assembling the right part of the differential equation.

The function takes as input one string (`function ltx(x)`)
Step 1. Replace "`*`" character (sign) with " `\cdot` " string (TeX encoding for multiplication)
```
y=str_replace("*"," \cdot",x);
```
Step 2. Split the fractions into nominator and denominator (`z=explode("/",y)`)
Step 3. If the split leads to only one element, then it is the nominator (and it should be returned as is) else go to the next step (`if(count(z)<2)return(z[0])`)
Step 4.
  4.1. Find first position of the "`+`" sign (`i1=strpos(z[0],"+")`)
  4.2. Find first position of the "`-`" sign (`i2=strpos(z[0],"-")`)
  4.3. Decide whatever comes first
```
i3===FALSE;if(i2===FALSE)i3=i1;if(i1===FALSE)i3=i2;
if((!(i1===FALSE))&&(!(i2===FALSE)))if(i2>i1)i3=i2;else
i3=i1;
```

Table 2
Gradients after pressure (p), volume (V) and temperature (T) of an unknown function (f): the case of total energy ($f = E$).

	Variable		$\left.\frac{\partial f}{\partial x}\right|_{z=ct} = \left.\frac{\partial f}{\partial x}\right|_{y=ct} + \left.\frac{\partial f}{\partial y}\right|_{x=ct} \cdot \left.\frac{\partial y}{\partial x}\right|_{z=ct}$	$\left.\frac{\partial f}{\partial z}\right|_{x=ct} = \left.\frac{\partial f}{\partial y}\right|_{x=ct} \cdot \left.\frac{\partial y}{\partial z}\right|_{x=ct}$	Order*
z	x	y	fxz = fxy + fyx·yxz	fzx = fyx·yzx	f = E
p	T	V	fpV = fpT + fTp·TpV	fVp = fTp·TVp	6
p	V	T	fpT = fpV + fVp·VpT	fTp = fVp·VTp	5
T	p	V	fTV = fTp + fpT·pTV	fVT = fpT·pVT	2
T	V	p	fTp = fTV + fVT·VTp	fpT = fVT·VpT	4
V	p	T	fVT = fVp + fpV·pVT	fTV = fpV·pTV	1
V	T	p	fVp = fVT + fTV·TVp	fpV = fTV·TpV	3

*Order identifies the possible order applied to identify the terms starting from the known toward unknowns using the relations between partial derivatives. Note that the derivatives in the top of the second and third columns are true only for functions that fulfil the following conditions: $f = f(x,y)$ and $y = y(x,z)$.