Saturated Pseudoadiabats—A Noniterative Approximation

ATOOSSA BAKHSHAII AND ROLAND STULL

Department of Earth, Ocean and Atmospheric Sciences, The University of British Columbia, Vancouver, Canada

(Manuscript received 7 March 2012, in final form 18 June 2012)

ABSTRACT

Two noniterative approximations are presented for saturated pseudoadiabats (also known as moist adiabats). One approximation determines which moist adiabat passes through a point of known pressure and temperature, such as through the lifting condensation level on a skew T or tephigram. The other approximation determines the air temperature at any pressure along a known moist adiabat, such as the final temperature of a rising cloudy air parcel. The method used to create these statistical regressions is a relatively new variant of genetic programming called gene-expression programming. The correlation coefficient between the resulting noniterative approximations and the iterated data such as plotted on thermodynamic diagrams is over 99.97%. The mean absolute error is 0.28°C, and the root mean square error is 0.44 within a thermodynamic domain bounded by $-30° < \theta_w \le 40°$C, $P > 20$ kPa, and $-60° \le T \le 40°$C, where θ_w, P, and T are wet-bulb potential temperature, pressure, and air temperature.

1. Introduction and overview of pseudoadiabatic theory

a. Dry adiabats

In a thermodynamic diagram such as the skew T–log$p$ in Fig. 1, the ''dry'' adiabat is described by a noniterative equation [Emanuel 1994, Eq. (4.2.11)]:

$$T_f = T_i(P_f/P_i)^{b(R_d/C_{\rm pd})}. \tag{1}$$

Noniterative means the absolute temperature T_f at any final pressure P_f can be solved directly knowing the initial pressure and absolute temperature (P_i, T_i). Here R_d is the gas constant for dry air, $C_{\rm pd}$ is the specific heat at constant pressure for dry air, and their dimensionless ratio is $R_d/C_{\rm pd} = 0.285\,71$. The word ''dry'' means unsaturated humid air. For the special case of $P_f = 100$ kPa, T_f is defined as the potential temperature θ, and this theta can be used as a label for the dry adiabat.

Factor b in Eq. (1) explains the variations of C_p and R with humidity. Emanuel [1994, Eq. (4.2.11)] gives dimensionless factor b as

$$b = [1 + (r/\varepsilon)]/[1 + (r/c)] \approx (1 - 0.24r), \tag{2}$$

where r ($\text{kg}_{\text{watervapor}}\ \text{kg}^{-1}_{\text{dryair}}$) is the water-vapor mixing ratio, $\varepsilon = R_d/R_v = 0.622\ \text{kg}_{\text{watervapor}}\ \text{kg}^{-1}_{\text{dryair}}$ is the ratio of ideal gas constants for dry air and water vapor, and $c = C_{\rm pd}/C_{\rm pv} \approx 0.5427\ \text{kg}_{\text{watervapor}}\ \text{kg}^{-1}_{\text{dryair}}$ is the ratio of specific heats at constant pressure for dry air and water vapor at 0°C. In the atmosphere r is usually less than $0.04\ \text{kg}_{\text{watervapor}}\ \text{kg}^{-1}_{\text{dryair}}$, hence $b \approx 1$ with an error of less than 1% (Bolton 1980).

From Eq. (1), the slope of the dry adiabat in pressure coordinates is

$$\partial T/\partial P = bR_dT/(C_{\rm pd}P). \tag{3}$$

In height coordinates, this vertical temperature gradient [Emanuel 1994, Eq. (4.7.4)] is

$$\partial T/\partial z = -(g/C_{\rm pd})b(T/T_v) \equiv -\Gamma_d, \tag{4}$$

where $g = 9.8\ \text{m s}^{-1}$ is gravitational acceleration magnitude near the earth's surface, and the virtual absolute temperature is $T_v = T\{[1 + (r/\varepsilon)]/(1 + r)\} \approx T(1 + 0.608r)$. To good approximation, the dry-adiabatic lapse rate is $\Gamma_d \approx (g/C_{\rm pd}) \approx 9.76°\text{C km}^{-1}$.

b. Saturated pseudoadiabats

Unfortunately, the saturated pseudoadiabat (also sometimes called the wet adiabat or moist adiabat) does

Corresponding author address: Atoossa Bakhshaii, Dept. of Earth, Ocean and Atmospheric Sciences, The University of British Columbia, 6339 Stores Rd., Vancouver BC V6T 1Z4, Canada.
E-mail: abakhsha@eos.ubc.ca

DOI: 10.1175/JAMC-D-12-062.1

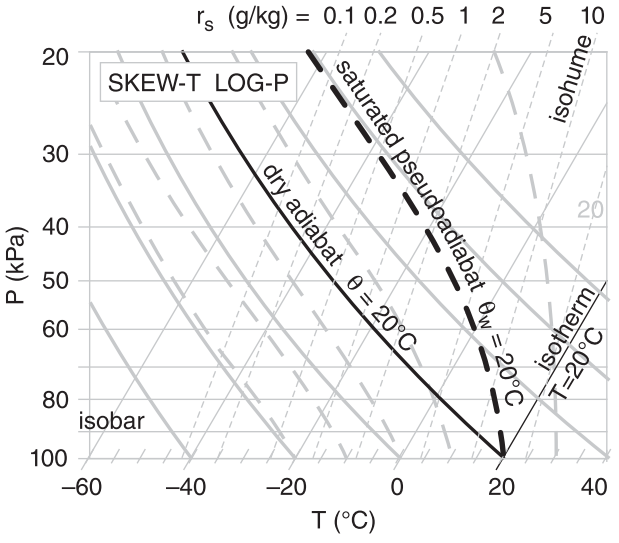

FIG. 1. Skew T–log$p$ thermodynamic diagram, with sample isopleth types highlighted in black and labeled. The thick dashed gray lines are a field of saturated pseudoadiabats.

not have such a direct definition. Instead, the slope $\partial T/\partial z$ or $\partial T/\partial P$ of the saturated pseudoadiabat can be found from conservation of moist entropy as a function of temperature and saturated mixing ratio r_s, which itself is a nonlinear function of T and P along that adiabat. The ideal gas constant R and the specific heat of air at constant pressure C_p are also functions of r_s. Hence, the equation for the pseudoadiabat must be iterated in small steps of ΔP from an initial (P_i, T_i) to find the temperature T_f at final pressure P_f. Such iteration can be computationally expensive.

For a pseudoadiabatic process, where all liquid water is assumed to fall out as soon as it forms, Emanuel [1994, Eqs. (4.7.3) and (4.7.5)] gives the saturated lapse rate ($\Gamma_s = -\partial T/\partial z$) as

$$\frac{\Gamma_s}{\Gamma_d} = \frac{1 + \dfrac{L_v r_s}{R_d T}}{1 + \dfrac{L_v^2 r_s \varepsilon b}{R_d C_{\mathrm{pd}} T^2}}, \tag{5}$$

where Γ_d is from Eq. (4), and the saturation mixing ratio is

$$r_s = \varepsilon e_s/(P - e_s). \tag{6}$$

The saturation vapor pressure e_s can be found from the Clausius–Clapeyron equation, from Tetens's equation, or from Bolton's equation [Bolton 1980, Eq. (10)]. Emanuel (1994) uses Bolton's equation:

$$e_s = e_o \exp\left[\frac{17.67(T - 273.15\ \mathrm{K})}{T - 29.65\ \mathrm{K}}\right], \tag{7}$$

for T in kelvins, and where the reference vapor pressure is $e_o = 0.6112$ kPa at 0°C. The other parameters in Eq. (5) as given by Bolton (1980) are the latent heat of vaporization L_v ($\approx 2.501 \times 10^6$ J kg^{-1} at 0°C), the gas constant for dry air $R_d = 287.053$ J K^{-1} kg^{-1}, and the specific heat at constant pressure for dry air C_{pd} (≈ 1004 J K^{-1} kg^{-1} at 0°C).

The hydrostatic equation and the ideal gas law can be used to rewrite Eq. (5) in terms of pressure:

$$\frac{\partial T}{\partial P} = \left(\frac{b}{P}\right)\frac{R_d T + L_v r_s}{C_{\mathrm{pd}} + \dfrac{L_v^2 r_s \varepsilon b}{R_d T^2}}. \tag{8}$$

Because pressure decreases as height increases, Eq. (8) gives a vertical temperature gradient of the proper sign. Equations (5) and (8) are similar to ones presented by Bohren and Albrecht [1998, Eq. (6.111)]. Further simplifications occur by using $b \approx 1$.

Thus, the iterative solution proceeds as follows. For any starting $[P_1, T(P_1)]$ such as at an initial point $(P_i, T_i) = (100\ \mathrm{kPa}, \theta_w)$ on the saturated adiabat, solve Eq. (7) for e_s, then solve Eq. (6) for r_s, then solve Eq. (2) for b (using r_s instead of r) or use the approximation $b \approx 1$, then solve Eq. (8) for $\partial T/\partial P$. Next, assume that the saturated adiabat is approximately linear over a very small increment of pressure $\Delta P = P_2 - P_1$, giving $T(P_2) \approx T(P_1) + (P_2 - P_1)\partial T/\partial P$. This new temperature and pressure can be used as input to the next small increment, with the process repeated until the final destination pressure P_f is reached.

If Eq. (8) is initialized at a nonreference altitude (i.e., for $P \neq 100$ kPa) and iterated to reach a final pressure of $P_f = 100$ kPa, then the final temperature T_f is defined as the web-bulb potential temperature θ_w, which can be used as a label for that saturated pseudoadiabat (Fig. 2). If Eq. (8) is iterated to the top of the atmosphere ($P_f = 0$ kPa) then its final potential temperature is defined as the equivalent potential temperature θ_e, which is an alternative label (Fig. 2). Bolton [1980, Eq. (40)] and Emanuel [1994, Eq. (4.7.10)] give the following empirical relationship between θ_e and θ_w:

$$\theta_e = \theta_w \exp\{r_{\mathrm{so}}(1 + c_1 r_{\mathrm{so}})[(c_2/\theta_w) - c_3]\}, \tag{9}$$

for r_{so} ($\mathrm{g}_{\mathrm{watervapor}}\ \mathrm{g}_{\mathrm{dryair}}^{-1}$) = saturation mixing ratio at $P = 100$ kPa and $T = \theta_w$, $c_1 = 0.81\ \mathrm{g}_{\mathrm{dryair}}\ \mathrm{g}_{\mathrm{watervapor}}^{-1}$, $c_2 = 3376$ K $\mathrm{g}_{\mathrm{dryair}}\ \mathrm{g}_{\mathrm{watervapor}}^{-1}$, and $c_3 = 2.54\ \mathrm{g}_{\mathrm{dryair}}\ \mathrm{g}_{\mathrm{watervapor}}^{-1}$. After some test iterations of Eq. (8) from $P_i = 100$ kPa

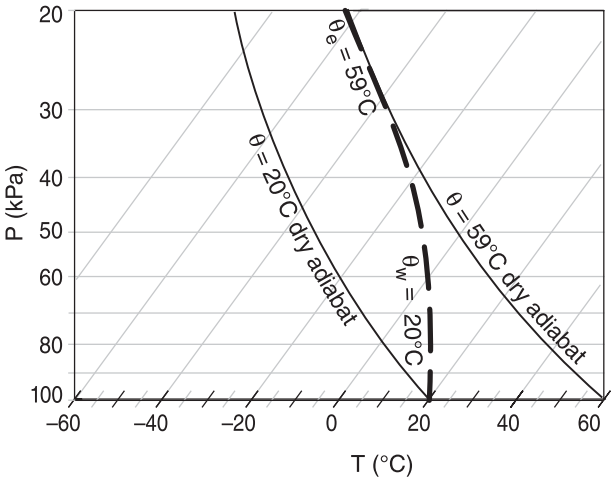

FIG. 2. Illustration of two methods to label saturated pseudoadiabats (thick dashed line). One is the potential temperature of the pseudoadiabat at a reference pressure of $P = 100$ kPa at the bottom of this diagram, which defines the wet-bulb potential temperature θ_w. The other is the potential temperature toward which the pseudoadiabat asymptotically approaches near the top of this diagram, which defines the equivalent potential temperature θ_e.

to $P_f = 10$ kPa with an increment of $\Delta P = 0.05$ kPa, it was found that the saturated pseudoadiabat to be asymptotic to a dry adiabat having $\theta = \theta_e$ given by Eq. (9).

c. Goal and motivation

The goal of this paper is to present noniterative approximations to the saturated pseudoadiabats. The motivation is to accelerate numerical weather prediction (NWP), where moist-adiabatic computations must be preformed in each grid column for each time step. As NWP models continue to achieve finer grid resolution and require larger numbers of smaller time steps, the number of required calculations increases exponentially. Computational savings by replacing repeated iterative computations for pseudoadiabats with direct computation could enable forecasts to finish sooner. Pseudoadiabatic computations provide important input to other NWP subroutines for buoyancy, convection, turbulence, cloud microphysics, and precipitation. Hence, NWP forecasts that finish sooner can provide earlier warnings of severe storms, heavy precipitation, wind and solar inputs to clean energy production, and other weather elements produced by NWP.

Section 2 examines differences among a few published thermodynamic diagrams and describes which data will be used for this study. Section 3 explains the methodology of symbolic regression used to find predictands from predictors. Section 4 gives the regression result for the wet-bulb potential temperature θ_w as a function of pressure and temperature (P, T). This identifies which specific pseudoadiabat (θ_w = constant) is to be used for saturated ascent or descent. Section 5 gives an expression for T as a function of θ_w and destination pressure P. Conclusions are in section 6.

2. Data

To create a noniterative relationship for the pseudoadiabat, one needs to regress the functions against training data of θ_w (P, T). Although these data could be found iteratively from the pseudoadiabatic theory of section 1, they could alternately be found from data corresponding to traditional printed thermodynamic diagrams. Either approach could yield useful input data—for the present research, fitting the traditional thermodynamic diagrams was chosen. These diagrams, and their differences from the theory of section 1, are explained next.

a. Diagram types and their operational implementations

Thermodynamic information including dry and saturated pseudoadiabats can be plotted in a variety of formats: emagram, skew T–log$p$, tephigram, Stüve, θ–z, θ_w–r_T, and others (Hess 1959; Emanuel 1994; Ambaum 2010). Identical thermodynamic information can be plotted with identical accuracy on any of these diagrams.

Historically, different forecast centers chose to use different formats, which they printed on large-size paper for use in operational plotting of atmospheric soundings. For example, the skew T diagram is commonly used in the United States (DoD 1978; AWS 1990), the tephigram dominates in many British commonwealth countries (Ambaum 2011) including Canada (EC 1976), and the Stüve has been used in some Germanic countries and by some aviation authorities (ATAA 1942). These three printed diagrams are pseudoadiabatic diagrams, where all liquid water is assumed to fall out immediately. Printed soundings are rarely used any more except in education. Instead computer-generated soundings are freely available via the Internet in a wide variety of formats. Nonetheless, many meteorologists picture the high-resolution printed diagrams in their minds as a reference when they view and interpret the coarser-resolution computer-generated soundings.

However, there is slight disagreement among the printed diagrams listed above. Most agree with each other in the bottom half of the troposphere, but there is some divergence in values higher in the atmosphere for warm saturated pseudoadiabats. For example, following the $\theta_w = 30$°C saturated pseudoadiabat up to a pressure of $P = 20$ kPa yields temperatures of −33.3°, −32.6°, and −33.2°C on the U.S. Air Force (USAF)

skew T, Environment Canada (EC) tephigram, and Air Transport Association of America (ATAA) Stüve. Some of these differences might be related to drawing errors of the original master diagram, and some are likely due to small differences in values of thermodynamic parameters. Bohren and Albrecht (1998) demonstrate that the $\theta_w = 25$°C saturated adiabat can be 3°C warmer at $P = 20$ kPa than the corresponding pseudoadiabat, and the ice-saturated pseudoadiabat can be 2°C warmer than the water-saturated pseudoadiabat.

Given typical rawinsonde uncertainties on the order of 0.5°C near the top of the troposphere (Vaisala 2010) and larger errors in their spatial representativeness, the small differences between different diagrams and their associated assumptions are relatively negligible.

b. Data selection and preparation

Needed for this study is a data field (a 2D array) of θ_w (P, T) values as input to train and test our regressions. When Eq. (8) was used to generate these values, the resulting θ_w curves were found to deviate slightly from the curves that were printed in the skew T, tephigram, and Stüve diagrams that were cited earlier. This deviation had greater magnitude for larger values of r, and at higher altitudes. To reduce this deviation, we manually tested slight changes to both Eqs. (8) and (7), until by trial and error we found the following equations that gave a better empirical fit to the printed diagrams:

$$\frac{\partial T}{\partial P} = \left(\frac{1}{P}\right) \frac{R_d T + L_v r_s}{(C_{\rm pd} - C_w r_s) + \dfrac{L_v^2 r_s \varepsilon}{R_d T^2}}, \qquad (10)$$

with $L_v = 2.50 \times 10^6$ J kg^{-1}, $R_d = 287.04$ J K^{-1} kg^{-1}, $\varepsilon = 0.621\,97$ kg$_{\rm water}$ kg$_{\rm dryair}^{-1}$, $C_{\rm pd} = 1005$ J K^{-1} kg^{-1}, $C_w = 4218$ J K^{-1} kg^{-1} (the specific heat for liquid water), and where Eq. (6) is used for r_s. For the saturation vapor pressure in Eq. (6), Tetens's formula is used [Bolton 1980, Eq. (8)]:

$$e_s = e_o \exp[17.269\,39(T - T_o)/(T - 35.86)], \qquad (11)$$

which was modified to use T in kelvins and $T_o = 273.16$ K, and which uses $e_o = 0.610\,78$ kPa [empirically found to give better results in Eq. (10), but slightly different from Bolton's value of about $e_o = 0.611$ kPa]. For different starting temperatures at $P = 100$ kPa, these equations were iterated to 10 kPa to create the target data values.

For comparison, when Eqs. (8) and (10) are iterated in 0.2-kPa increments along the $\theta_w = 30$°C saturated adiabat from initial point (P_i, T_i) = (100 kPa, 30°C) to final point (P_f, T_f) = (20 kPa, T_f), the resulting final temperatures are −31.36°C for Eq. (8), −31.46°C for Eq. (8) with $b = 1$, and −33.07°C for Eq. (10). This last value [from Eq. (10)] is within the range of $-33.2° \le T_f \le -32.6$°C from the printed diagrams. While the deviation magnitude between Eqs. (8) and (10) was about 1.6°C for this very warm θ_w case, the deviations of the $\theta_w = 10$°C moist adiabat at $P = 50$ kPa were smaller—roughly 0.3°C magnitude. In summary, Eq. (10) was found to fit better the printed diagrams for the saturated pseudoadiabats, but the differences between Eqs. (8) and (10) are small for normal meteorological conditions.

Equation (10) is asymptotic to a different dry adiabat than is Eq. (8). For those moist adiabats that reach their asymptote at pressures in the range of $100 \le P \le 20$ kPa, the following two-parameter relationship between θ_e and θ_w was found for the thermodynamic diagrams studied here:

$$\theta_e = \theta_w \exp[r_{\rm so}(1 + c_1 r_{\rm so})(c_2/\theta_w)], \qquad (12)$$

for $r_{\rm so}$ (kg$_{\rm watervapor}$ kg$_{\rm dryair}^{-1}$) = saturation mixing ratio at $P = 100$ kPa and $T = \theta_w$, $c_1 = 0.81$ kg$_{\rm dryair}$ kg$_{\rm watervapor}^{-1}$, $c_2 = 2555$ K kg$_{\rm dryair}$ kg$_{\rm watervapor}^{-1}$, and where Tetens's formula [Eq. (11)] was used for saturation vapor pressure. Equation (12) is based on the lowest altitude at which the moist adiabat is parallel, or nearly parallel, to a dry adiabat. Although Eq. (12) is not used for our noniterative approximations, it is presented here because some readers may prefer to identify the saturated pseudoadiabat by θ_e rather than θ_w. For a thorough discussion of methods to calculate by θ_e from θ_w, including comparisons with a one-parameter form by Bryan (2008) and the three-parameter form of Bolton (1980), see Davies-Jones (2008, 2009).

3. Methodology

a. Two algorithms needed

When using a thermodynamic diagram or its computational equivalent, two operations are often needed with regard to saturated-adiabatic processes (i.e., for a cloudy air parcel). First, one needs to determine which saturated pseudoadiabat the air parcel follows from the initial state (P_i, T_i). Second, after following this pseudoadiabat to some final pressure P_f, one needs to know the final temperature T_f of the air parcel. Needed for the first operation is $\theta_w(P, T)$, while needed for the second is $T(P, \theta_w)$. These two operations are analogous to entering and exiting a highway, where the highway is the saturated pseudoadiabat (Fig. 3). Note, the mixing ratio is not included as an independent variable in the previous

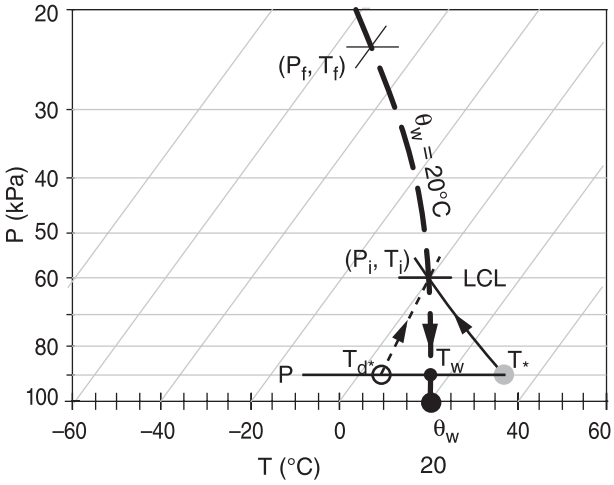

FIG. 3. Illustration of arbitrary initial (i) and final (f) points along an arbitrary saturated pseudoadiabat (thick dashed line) of wet-bulb potential temperature $\theta_w = 20$°C. Also shown are the wet-bulb temperature T_w of the initial unsaturated air parcel of temperature T_* and dewpoint T_{d^*}, and the corresponding LCL.

two lines because for saturation it is dependent on pressure and temperature.

Users of thermodynamic diagrams often determine the initial conditions for the first operation from the lifting condensation level (LCL) of a hypothetical rising unsaturated air parcel $(P_i, T_i) = (P_{\rm LCL}, T_{\rm LCL})$. Knowing the initial conditions (P_*, T_*, T_{d^*}) of an unsaturated air parcel where T_d is the dewpoint temperature, then the LCL conditions can be estimated from

$$P_{\rm LCL} \approx P_* \left[1 - a\left(\frac{T_* - T_{d^*}}{T_*}\right)\right]^{C_{\rm pd}/R_d}, \qquad (13)$$

where $a = 1.225$ is a dimensionless empirical constant (Stull 2000) and $C_{\rm pd}/R_d = 3.50$ (dimensionless). The $T_{\rm LCL}$ can then be solved from Eq. (1) by using dry-adiabatic initial state (P_*, T_*) and setting the final state to $(P_{\rm LCL}, T_{\rm LCL})$. Equation (13) is not part of the current study—indeed, other researchers might use more precise methods to find the LCL or other initial conditions to define θ_w. It is assumed that the user provides appropriate initial conditions (P_i, T_i).

Each operation $[\theta_w(P, T)$ or $T(P, \theta_w)]$ requires a separate regression, but both regressions are fit to the same dataset that is created iteratively with Eq. (10). If one of the regressions had resulted in a simple algorithm, then it could have been solved analytically (i.e., inverted) for the other operation without requiring a second regression. However, neither algorithm is simple, so both regressions are required.

An evolutionary regression method is used here, where not only the constants (or weights) in the algorithm are found, but also the functional form of the algorithm can vary (viz., it is not limited to any a priori algorithm form such as a polynomial or a neural network). This regression is performed using gene-expression programming.

b. Gene-expression programming

Gene-expression programming (GEP) is an evolutionary method that can be used to find a best-fit function to data. Recently devised by Ferreira (2006), GEP is an efficient variant of genetic programming and genetic algorithms, where candidate functions evolve via various forms of mutation and compete via a computational natural selection until the fittest candidate (with the lowest verification error) is found. GEP can explore a wide range of the function space to find a best fit, and can yield a nonlinear result that would not necessarily have been obvious if the function fit had been attempted manually.

The complexity of an algorithm that can be produced by GEP depends on the specification of which and how many basic functions (e.g., exp, sin, arctan) and operators (+, −, ∗, /) from which GEP is allowed to draw, and on the specification of the number of components (operators, functions, input predictor variables, and numerical constants) that are allowed to make up the algorithm. In the terminology of GEP, these components are represented by characters in a character string called a chromosome, where the chromosome consists of substrings called genes that are added together. The user designs the component complexity of GEP based on the complexity of the problem to be solved, and on the desire to create the simplest algorithm that adequately fits the target data. Such design usually requires some trial and error, which can build on the experience and recommendations of Ferreira (2006). The GEP specifications that were used for saturated pseudoadiabats are given in sections 4 and 5.

The focus of this paper is on the noniterative pseudoadiabat algorithm that results from using GEP, and not on the inner workings of GEP itself. For details on GEP, the reader is referred to Bakhshaii and Stull (2009, 2012) and Ferreira (2006). We used a commercial computer package called GeneXproTools (Gepsoft 2012) that can run on personal computers. Some recent meteorological applications of GEP include estimating ensemble-average precipitation in mountainous terrain from numerical forecasts (Bakhshaii and Stull 2009), forecasting minimum daily temperature as a model output statistic (Roebber 2010), approximating wet-bulb temperature from relative humidity and air temperature

(Stull 2011), and estimating electric load for a population of 4.5 million people (Bakhshaii and Stull 2012).

The reader is cautioned that the best-fit results that we found using GEP are just two realizations out of perhaps an infinite set of alternatives that could provide nearly identical goodness-of-fit. Such is the quasi-random nature of evolutionary programming. Namely, a different researcher could start with the same training data and the same GEP specifications, yet reach a different and equally good result.

4. Wet-bulb potential temperature from pressure and temperature: $\theta_w(P, T)$

a. GEP setup

GEP was able to find an algorithm to fit θ_w as a function of initial P and T. GEP was programmed to use a mutating population of 30 individuals that competed over thousands of generations for the best fitness. Each individual had a chromosome made of the sum of six genes, where each gene had a maximum size of 29 components (made up of combinations of the two predictor variables P and T, the operators described next, and mutable numerical weights). The six genes were simply added together to create the full chromosome. For this evolution, GEP was allowed to randomly select from the following set of operators and functions during initialization and for subsequent mutations: +, −, ∗, /, sqrt, exp, ln, square, cube, sin, cos, cube root, and arctan. GEP was set to use two weights per gene, where the value of each weight was allowed to mutate within the range from −10.0 to 10.0.

The following parameters were also used in GEP:

- single-point mutation at a rate of two mutations per chromosome per generation;
- inversion (reversing the order of components in a substring of the chromosome) at rate 0.1 (where the rate is the portion of the population that changes in each generation);
- insertion sequence transposition (moving a substring to a different location) at rate 0.1;
- root insertion sequence transposition (moving a substring to the front of a gene) at rate 0.1;
- gene transposition (moving a whole gene to the front of the chromosome) at rate 0.1;
- one-point recombination (swapping trailing substrings between two individuals in the population) at rate 0.3;
- two-point recombination (swapping interior substrings between two individuals) at rate 0.3;
- gene recombination (swapping whole genes between individuals) at rate 0.1.

These parameter values are the default values initialized by GEP, which have been shown by Ferreira (2006) to provide rapid computational evolution.

The computational selection of which individuals survive (some with mutation) into the next generation is based on the fitness f of each individual. This fitness is defined as $f = 1000\,(1 + \mathrm{RRSE})^{-1}$, where the root-relative square error is

$$\mathrm{RRSE} = \left\{ \left[\sum_{i=1}^{n} (E_i - O_i)^2 \right] \Big/ \left[\sum_{i=1}^{n} (\overline{O} - O_i)^2 \right] \right\}^{1/2}. \tag{14}$$

In this expression, E_i is the value (θ_w in our case) estimated by the individual algorithm for the ith data point, O_i is the corresponding target value, n is the total number of data points, and the overbar indicates a mean value. RRSE can be interpreted as the error variance relative to the variance of the target data around the mean $\overline{O}$. A perfectly fit individual has a fitness of 1000, while an unfit individual has $f = 0$.

The total target dataset consisted of values of $\theta_w(P, T)$ as calculated iteratively using Eq. (10) for air-parcel temperatures in the range of $-60° \le T \le 40°$C. These points were created for 66 separate saturated pseudoadiabat curves in the range of $-25° \le \theta_w \le 40°$C with θ_w increments of 1°C. To maintain accuracy, each θ_w curve was iterated, starting from $P = 100$ kPa, with a very small pressure increment of $\Delta P = 0.1$ kPa up to an altitude where $P = 10$ kPa. This led to over $N = 44\,000$ data points. However, because the pseudoadiabats vary smoothly within the domain, we were able to reduce the data volume by subsampling along the θ_w curves at constant increments of $\Delta P = 1$ kPa to get the target values O for training GEP. This reduced the data to $n = 4397$ points, which allows faster evolution by GEP. The domain of these points is the whole unshaded area in Fig. 4; namely, it includes many moist adiabats not sketched in that figure.

The viability of final candidate algorithms that were produced by GEP were tested against the full dataset ($n = N$). Such a procedure of testing against a larger dataset than is used for training is a method that helps to reduce overfitting (to avoid fitting the noise instead of the signal).

b. Results

The final result from GEP is the following noniterative algorithm for determining $\theta_w(P, T)$:

$$\theta_w = g_1 + g_2 + g_3 + g_4 + g_5 + g_6, \tag{15a}$$

where the six genes are defined as follows:

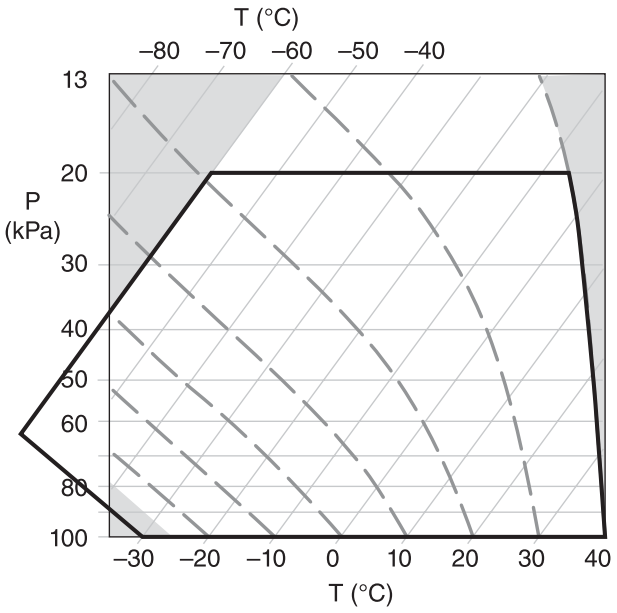

FIG. 4. Nonshaded region of the skew T diagram shows most of the subdomain from which data points were taken to train GEP to find $\theta_w(P, T)$. Thick black line outlines the subdomain used to calculate verification scores.

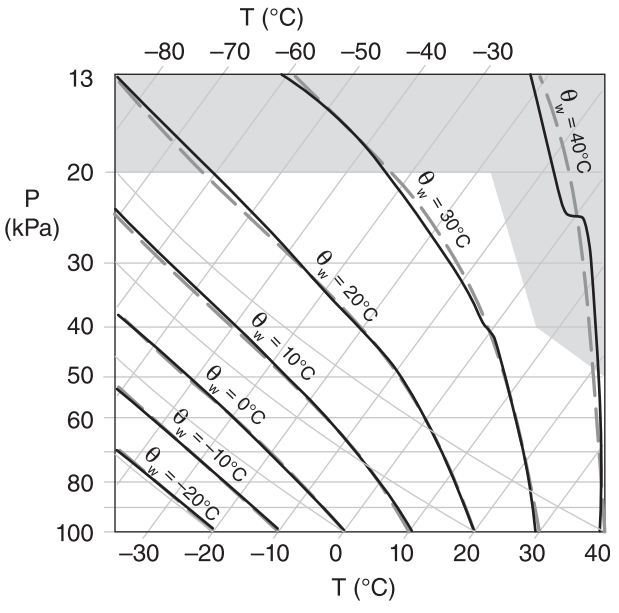

FIG. 5. Wet-bulb potential temperature θ_w as a function of temperature T and pressure P. The target pseudoadiabats [iterative Eq. (10)] are plotted as thick gray dashed curves, and the approximations [noniterative Eq. (15)] are plotted as thick black curves. Several dry adiabats are plotted for reference as thin gray lines sloping up to the left. For the USAF published skew T diagram, pseudoadiabats are provided only within the unshaded subdomain.

$$g_1 = \arctan\{-0.014\,174\,8[P^{1/2}(8.114\,196 + T) + 65.8402]\}, \quad (15b)$$

$$g_2 = (69.2840 + P^{1/2})^{1/2} + [6.558\,563 + (8.3237/P)]^2, \quad (15c)$$

$$g_3 = \exp(17.850\,425/P)\sin[0.0510(T - P)], \quad (15d)$$

$$g_4 = 0.007\,404\,25(T - 23.9263)P, \quad (15e)$$

$$g_5 = -0.355\,695[0.5997 + P - T + \arctan(T)], \quad \text{and} \quad (15f)$$

$$g_6 = 0.357\,635[0.0922 + \arctan(T)]\sin[(3.877\,869 + P)^{1/2}]. \quad (15g)$$

In these equations, θ_w and T must have units of degrees Celsius, P must have units of kilopascals, and the trigonometric functions treat their arguments as if they are in radians.

Figure 5 shows a comparison of the regressed (noniterative) approximation for θ_w calculated using Eq. (15) versus the target from the iterative solution to Eq. (10). For comparison of plotting domains, the white background in Fig. 5 shows the subdomain for which pseudoadiabats are provided on the published USAF skew T (DoD 1978; AWS 1990). Although our noniterative approximation has a kink around 0°C at high wet-bulb potential temperatures, this is outside the domain normally used. Figure 6 shows the corresponding θ_w errors as a function of P and T. These errors are on the same order of magnitude as the errors caused by not knowing whether the real atmospheric process is adiabatic or pseudoadiabatic. The shaded subdomains in these figures show where errors were greater than 1°C. Extrapolation of Eq. (15) beyond the region for which it was trained can give very large errors and is not recommended.

Verification was done using 44 824 data points distributed within the skew T subdomain outlined with the thick black line in Fig. 4. These were the same points found iteratively for a range of θ_w and P, as was described in section 4a, excluding all points at pressures less than 20 kPa. The verification results are mean absolute error (MAE) = 0.28°C, root mean square error (RMSE) = 0.44°C, root-relative square error (RRSE) = 0.02, and $r^2 = 0.9995$ between the iterated target points and the noniterative approximation, where r is the Pearson correlation coefficient.

5. Temperature from pressure and wet-bulb potential temperature: $T(P, \theta_w)$

a. Setup

While GEP was able to find a single algorithm to fit T as a function of θ_w and final P, this algorithm had unacceptably large temperature errors (≥1°C over many

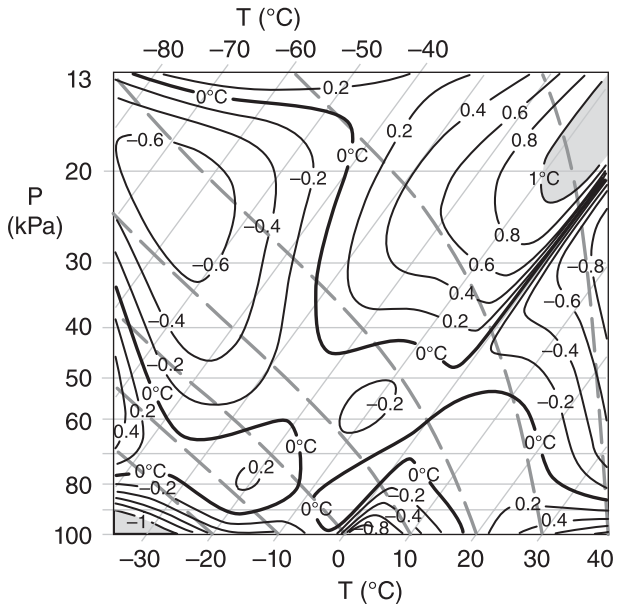

FIG. 6. Black curves show errors ($\theta_{w,\text{noniterative}} - \theta_{w,\text{iteratedtarget}}$) of wet-bulb potential temperature (°C) as a function of temperature T and pressure P, corresponding to Fig. 5. The target pseudoadiabats [iterative Eq. (10)] are plotted as gray dashed curves, labeled at the bottom axis. Error magnitudes > 1°C are shaded.

portions of the thermodynamic diagram). However, by splitting the problem into three subdomains (cold: $-30° < \theta_w \le 4°C$; warm: $4° < \theta_w \le 21°C$; and hot: $21° < \theta_w < 45°C$), we were able to find separate regressions for each subdomain with errors mostly less than 0.5°C. Target data points were again calculated from Eq. (10) to span this expanded domain. After experimenting to find the smallest chromosomes and function lists that gave an acceptable fit, we arrived at the following GEP specifications.

For the cold θ_w subdomain the chromosome was the sum of six genes, each gene with a maximum of 29 components. The set of operators and functions available for random initialization and subsequent mutations were +, −, *, /, sqrt, exp, ln, square, cube, sin, cos, cube root, arctan, logistic, and the numerical constants 0, 1, e, and π. There were 2015 target data points in this subdomain.

For the warm θ_w subdomain the chromosome was the sum of seven genes, each gene with a maximum of 29 components. The set of operators and functions available for random initialization and subsequent mutations were +, −, *, /, sqrt, exp, ln, square, cube, sin, cos, cube root, arctan, inverse, log, and fourth root. In this subdomain there were 2124 target data points.

For the hot θ_w subdomain the chromosome was the sum of seven genes, each gene with a maximum of 29 components. The set of operators and functions available for random initialization and subsequent mutations were +, −, *, /, sqrt, exp, ln, square, cube, sin, cos, cube root, arctan, fourth root, fifth root, min, max, logistic, and the numerical constants 0, 1, e, and π. In this subdomain there were 2604 target data points.

For each of the subdomains listed above, we used the same population size, mutation rates and mutable real-number weights as was described in section 4a.

b. Results

For the cold subdomain of $-30° < \theta_w \le 4°C$, the noniterative algorithm for determining T (°C) from θ_w (°C) and P (kPa) is

$$T = g_1 + g_2 + g_3 + g_4 + g_5 + g_6, \tag{16a}$$

where the six genes are

$$g_1 = -20.3313 - 0.0253P, \tag{16b}$$

$$g_2 = \sin[(\theta_w + P)^{1/2}] + (\theta_w/P) + P - 2.8565, \tag{16c}$$

$$g_3 = \cos\{19.6836 + [1 + \exp(-\theta_w)]^{-1/3} + P/15.0252\}, \tag{16d}$$

$$g_4 = 4.4653\sin(P^{1/2}) - 71.9358, \tag{16e}$$

$$g_5 = \{\exp[\theta_w - 2.718\,28\cos(P/18.5219)]\}^{1/6}, \quad \text{and} \tag{16f}$$

$$g_6 = \theta_w - \sin\{[P + \theta_w + \arctan(\theta_w) + 6.6165]^{1/2}\}. \tag{16g}$$

For the warm subdomain of $4° < \theta_w \le 21°C$, the noniterative algorithm for determining T (°C) from θ_w (°C) and P (kPa) is

$$T = g_1 + g_2 + g_3 + g_4 + g_5 + g_6 + g_7, \tag{17a}$$

where the seven genes are

$$g_1 = -9.6285 + \cos(\ln\langle\arctan\{\arctan[\exp(-9.2121\theta_w/P)]\}\rangle), \tag{17b}$$

$$g_2 = \theta_w - (19.9563/P)\arctan(\theta_w) + \theta_w^2/(5.471\,62P), \tag{17c}$$

$$g_3 = \sin[\ln(8P^3)]\ln(2P^{3/2}), \tag{17d}$$

$$g_4 = \theta_w + (P\theta_w - P + \theta_w)/(P - 190.2578), \tag{17e}$$

$$g_5 = P - (P - 383.0292)/(15.4014P - P^2), \tag{17f}$$

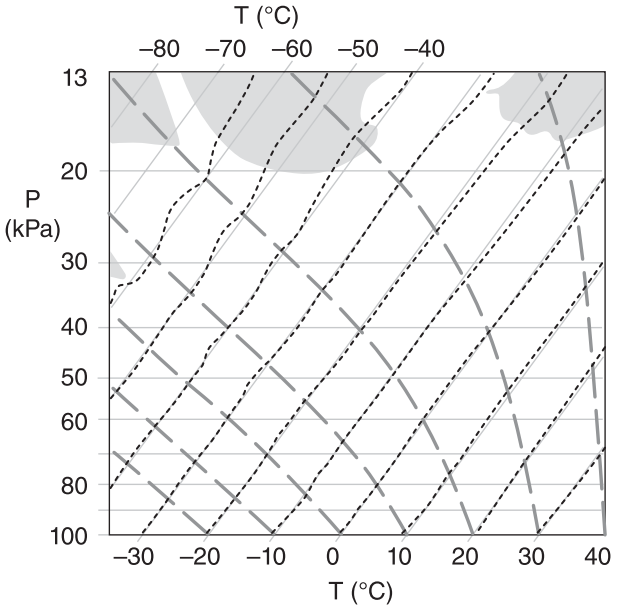

FIG. 7. Air temperature T as a function of wet-bulb potential temperature θ_w (thick dashed gray curves) and pressure P (thin gray horizontal lines). The target isotherms [iterative Eq. (10)] are plotted as thin diagonal gray lines, and the approximations [noniterative Eqs. (16)–(18)] are plotted as dotted black lines. Error magnitudes > 2°C are shaded.

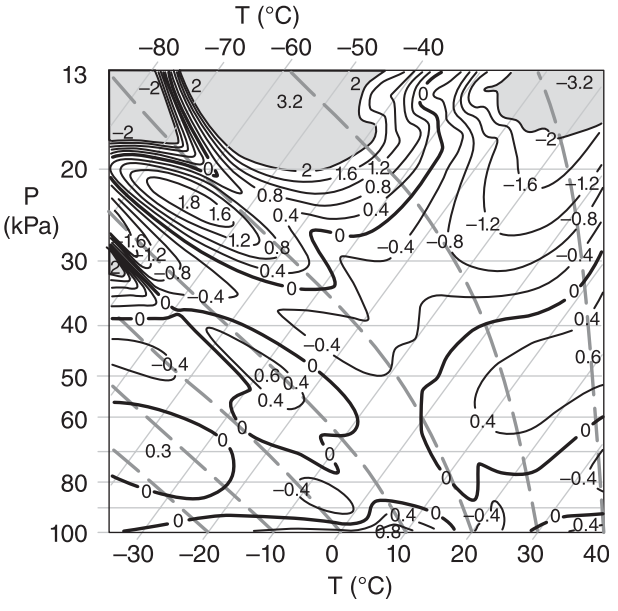

FIG. 8. Black curves show errors ($T_{\text{noniterative}} - T_{\text{iteratedtarget}}$) of air temperature (°C) as a function of wet-bulb potential temperature θ_w (thick dashed gray curves) and pressure P (thin gray horizontal lines). These errors correspond to the curves in Fig. 7. Error magnitudes > 2°C are shaded.

$$g_6 = (1/3)\ln(339.0316 - P) + \arctan(\theta_w - P + 95.9839), \tag{17g}$$

and

$$g_7 = -\ln(P)(298.2909 + 16.5109P)/(P - 2.2183). \tag{17h}$$

For the hot subdomain of $21° < \theta_w < 45$°C, the noniterative algorithm for determining T (°C) from θ_w (°C) and P (kPa) is

$$T = g_1 + g_2 + g_3 + g_4 + g_5 + g_6 + g_7, \tag{18a}$$

where the seven genes are

$$g_1 = 0.3919\theta_w^{7/3}[P(P + 15.8148)]^{-1}, \tag{18b}$$

$$g_2 = [19.9724 + (797.7921/P)]\sin(-19.9724/\theta_w), \tag{18c}$$

$$g_3 = \{\ln(-3.927\,765 + \theta_w + P)\cos[\ln(\theta_w + P)]\}^3, \tag{18d}$$

$$g_4 = \langle\exp\{[\theta_w + (1 + e^{-P})^{-1}]^{1/2} - 1.5603\}\rangle^{1/2}, \tag{18e}$$

$$g_5 = (P + \theta_w)^{1/2}\exp\{\arctan[(P + \theta_w)/7.9081]\}, \tag{18f}$$

$$g_6 = \{(P/\theta_w^2)\min[9.6112, (P - \theta_w)]\} - 13.7300, \tag{18g}$$

and

$$g_7 = \sin\{\sin^3[\min(P, 17.3170)] - P^{1/2} + 25.5113/\theta_w\}. \tag{18h}$$

As before, the trig functions in Eqs. (16)–(18) treat their arguments as if they are in radians, and the ''min'' function returns the minimum of two arguments. Sample calculations with Eqs. (15)–(18) are given in the appendix.

Figure 7 shows a comparison of the regressed (noniterative) approximations for T calculated using Eqs. (16)–(18) versus the target from the iterative solution to Eq. (10). Figure 8 shows the corresponding T errors as a function of P and θ_w. Verification was done using 44 824 data points distributed uniformly within the skew T subdomain outlined with the thick black line in Fig. 4. The verification results were MAE = 0.27°C, RMSE = 0.36°C, RRSE = 0.01, and r^2 = 0.9998. The shaded subdomains in Figs. 7 and 8 show where error magnitudes exceed 2°C. Extrapolation of Eqs. (16)–(18) into and beyond these shaded regions is not appropriate, and can give very large errors.

Whenever domains are broken into subdomains for a separate solution, there is concern that values could be discontinuous at the subdomain boundaries. Small T discontinuities are indeed observed at the boundaries ($\theta_w = 4°$ and 21°C) in Fig. 7. The corresponding T errors near those boundaries are nonetheless small—on the order of 0.5°C or less, except for a very small region near

(P, T) = (100 kPa, 4°C). To help minimize these errors, we used overlapping subdomains (θ_w = from −30° to +5°C; from −5° to 25°C; and from 15° to 45°C) when training each GEP regression. However, when using Eqs. (16)–(18), one need not worry about the overlap; that is, one can use the equations for the nonoverlapping ranges listed at the start of each of those equations, the results of which are plotted in Fig. 7. For typical thermodynamic calculations following a saturated air parcel moving adiabatically, recall that θ_w is constant, so the air parcel would not normally cross a subdomain boundary.

6. Discussion and conclusions

Two noniterative approximations were presented for saturated pseudoadiabats (also known as moist adiabats). One approximation [Eq. (15)] allows one to determine which moist adiabat passes through a point of known pressure and temperature, such as is drawn on a tephigram, skew T diagram, or other thermodynamic diagram. The other approximation [Eqs. (16)–(18)] allows one to determine the air temperature at any pressure along a known moist adiabat. The correlation coefficient r between the approximations and the thermodynamic diagram data is over 99.7%. The difference between the noniterative approximation and the corresponding iterated target values is less than about 0.5°C over a majority of the skew T subdomain that is outlined with the thick black line in Fig. 4. The distribution of errors is plotted in Figs. 6 and 8.

These approximations are based on a statistical regression, not on first principles. They apply only over the domains indicated on Figs. 6 and 8, and would give erroneous results if extrapolated outside that domain. The approximations given as Eqs. (15)–(18) are not the only ones possible. The random nature of evolutionary programming means that a large number of similarly accurate approximations could likely be found.

Although iteration is not needed when using Eqs. (15)–(18) to determine the thermodynamic state of a rising saturated air parcel, some meteorological applications require iteration nonetheless. Examples are buoyancy or stability calculations. Iteration is required for these cases because the environment surrounding the air parcel also changes as the parcel rises. Nonetheless, Eqs. (15)–(18) allow one to take larger vertical steps such as might correspond to significant levels in a sounding, rather than taking the smaller steps that would have been needed to accurately follow a saturated adiabat using iterative equations.

Acknowledgments. Funding for this research was provided by the Canadian Natural Science and Engineering Research Council, the Canadian Foundation for Climate and Atmospheric Science, and BC Hydro.

APPENDIX

Sample Calculations

While Eqs. (15)–(18) look complicated, they are simple to program or to enter in a spreadsheet. To allow readers to check their own code, sample calculations are given here.

Given an air parcel at its LCL of P = 80 kPa (i.e., 800 hPa) with temperature of T = 9°C, identify which saturated pseudoadiabat goes through this point. Equations (15b)–(15g) give $g_1 = -1.26$, $g_2 = 53.24$, $g_3 = 0.58$, $g_4 = -8.84$, $g_5 = -25.99$, and $g_6 = 0.15$. Summing these in Eq. (15a) gives θ_w = 17.9°C, which is very close to the skew T and tephigram values of slightly less than 18°C.

For cold θ_w air, given a saturated air parcel that follows the θ_w = −10°C saturated pseudoadiabat up to an altitude where the pressure is P = 70 kPa (i.e., 700 hPa), find its air temperature. Equations (16b)–(16g) give $g_1 = -22.10$, $g_2 = 67.99$, $g_3 = 0.73$, $g_4 = -68.04$, $g_5 = 0.27$, and $g_6 = -10.98$. Summing these in Eq. (16a) gives T = −32.1°C, which is very close to the skew T and tephigram values of −32.2°C.

For medium θ_w air, given a saturated air parcel that follows the θ_w = 16°C saturated pseudoadiabat up to an altitude where the pressure is P = 40 kPa (i.e., 400 hPa), find its air temperature. Equations (17b)–(17h) give $g_1 = -10.5$, $g_2 = 16.4$, $g_3 = 3.4$, $g_4 = 11.9$, $g_5 = 39.7$, $g_6 = 3.5$, and $g_7 = -93.6$. Summing these in Eq. (17a) gives T = −29.3°C, which is close to the skew T value of −28.6°C and the tephigram value of −28.2°C.

For hot θ_w air, given a saturated air parcel that follows the θ_w = 28°C saturated pseudoadiabat up to an altitude where the pressure is P = 25 kPa (i.e., 250 hPa), find its air temperature. Equations (18b)–(18h) give $g_1 = 0.9$, $g_2 = -33.9$, $g_3 = -18.2$, $g_4 = 6.8$, $g_5 = 30.2$, $g_6 = -13.8$, and $g_7 = 0.9$. Summing these in Eq. (18a) gives T = −27.2°C, which is close to the skew T value of −27.1°C and the tephigram value of −26.5°C.

REFERENCES

Ambaum, M. H. P., 2010: *Thermal Physics of the Atmosphere.* Wiley-Blackwell, 239 pp.

——, cited 2011: Tephigram. University of Reading, United Kingdom. [Available online at http://www.met.reading.ac.uk/~sws97mha/Tephigram/.]

ATAA, 1942: Pseudo-adiabatic diagram. Meteorological Committee Chart VII, Air Transport Association of America, 1 p.

AWS, 1990: The use of the skew T, log P diagram in analysis and forecasting. Air Weather Service Doc. AWS/TR-79/006, 159 pp.

Bakhshaii, A., and R. Stull, 2009: Deterministic ensemble forecasts using gene-expression programming. *Wea. Forecasting,* **24,** 1431–1451.

——, and ——, 2012: Electric load forecasting for western Canada: A comparison of two non-linear methods. *Atmos.–Ocean,* **50,** 352–363.

Bohren, C. F., and B. A. Albrecht, 1998: *Atmospheric Thermodynamics*. Oxford University Press, 402 pp.

Bolton, D., 1980: The computation of equivalent potential temperature. *Mon. Wea. Rev.,* **108,** 1046–1053.

Bryan, G. H., 2008: On the computation of pseudoadiabatic entropy and equivalent potential temperature. *Mon. Wea. Rev.,* **136,** 5239–5245.

Davies-Jones, R., 2008: An efficient and accurate method for computing the wet-bulb temperature along pseudoadiabats. *Mon. Wea. Rev.,* **136,** 2764–2785.

——, 2009: On formulas for equivalent potential temperature. *Mon. Wea. Rev.,* **137,** 3137–3148.

DoD, 1978: USAF skew *T*, log *P* diagram. U.S. Department of Defense Form DoD-WPC 9–16-1, 1 p.

EC, 1976: Tephigram. Environment Canada, Atmospheric Environment Service Form 0063-2394 1976, 1 p.

Emanuel, K. A., 1994: *Atmospheric Convection*. Oxford University Press, 580 pp.

Ferreira, C., 2006: *Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence*. 2nd ed. Springer, 478 pp.

Gepsoft, cited 2012: GeneXproTools. Gepsoft, Ltd., Bristol, United Kingdom. [Available online at http://www.gepsoft.com/.]

Hess, S. L., 1959: *Introduction to Theoretical Meteorology*. Henry Holt and Co., 362 pp.

Roebber, P. J., 2010: Seeking consensus: A new approach. *Mon. Wea. Rev.,* **138,** 4402–4414.

Stull, R., 2000: *Meteorology for Scientists and Engineers*. 2nd ed. Cengage, 502 pp.

——, 2011: Wet-bulb temperature from relative humidity and air temperature. *J. Appl. Meteor. Climatol.,* **50,** 2267–2269.

Vaisala, cited 2010: Datasheet for Vaisala radiosonde RS92-SGP. Vaisala, 2 pp. [Available online at http://www.vaisala.com/Vaisala%20Documents/Brochures%20and%20Datasheets/RS92SGP-Datasheet-B210358EN-E-LoRes.pdf.]