

A Formulation for the Relative Permittivity of Water and Steam to High Temperatures and Pressures Evolved Using Genetic Programming

*Sergey V. Fogelson and Walter D. Potter**

Artificial Intelligence Center

111 Boyd GSRC, Brooks Drive

University of Georgia

Athens, GA, 30602-7404

USA

potter@uga.edu

Author Bios and Photos (8/07)

The Journal of Physical Chemistry

“A Formulation for the Relative Permittivity of Water and Steam to High Temperatures and Pressures Evolved Using Genetic Programming”

Sergey V. Fogelson:

Sergey V. Fogelson graduated with an MS degree in Artificial Intelligence from the Artificial Intelligence Center at the University of Georgia, Athens, Georgia in July 2007, and a BA (with high honors) in Cognitive Science from the Artificial Intelligence Center in the same university also in July 2007. His current plans are to take some time off from academics to hike the Appalachian Trail before returning to complete his PhD degree in Computational Neuroscience and Neuroinformatics.

Walter D. Potter:

Walter D. Potter is Professor of Computer Science and Director of the Artificial Intelligence Center at the University of Georgia in Athens, Georgia. His research interests include knowledge-based systems, genetic algorithms, expert database systems, and advanced information system design. He has worked, as a research fellow, with the Oak Ridge National Laboratory, the US Army Research Institute, and the Westinghouse Savannah River Company. More recently, Dr. Potter has been involved with the USDA Forest Service doing research on ecosystem decision support systems. Dr. Potter received the BS degree from the University of Tennessee and the MS and PhD degrees from the University of South Carolina.



Sergey V. Fogelson



Walter D. Potter

Abstract The relative permittivity (or static dielectric constant) of water and steam has been experimentally obtained from a relatively wide range of temperatures and pressures. A suite of functions for predicting the relative permittivity of water and steam in four distinct thermodynamic regions is evolved using genetic programming. A data set comprised of all of the most accurate relative permittivity values, along with temperature, pressure, and density values from the entire experimentally obtained range of these values, is used for this task. The accuracy of these functions is evaluated by comparing the values for the relative permittivity calculated using the evolved function and the values calculated using the latest formulation of Fernandez et al, to the aforementioned data set. In all regions, the newly evolved function performs comparably to or better than the most current formulation in terms of difference between calculated and experimental values for the dielectric constant. This research heralds the first successful application of artificial evolutionary techniques to relative permittivity prediction in physical chemistry.

Keywords: genetic programming, relative permittivity, thermodynamic properties

INTRODUCTION

The relative permittivity (or static dielectric constant) of water and steam, ϵ_r , has been experimentally determined for a relatively wide range of temperatures and pressures. The relative permittivity is an important indicator of the solvent behavior of water in a variety of biological (cell membrane electrophysiology, intracellular biochemical processes) and geophysical/industrial (geochemical high temperature, high pressure processes in deep sea vents and in industrial processing plants) settings¹. Over the years, many researchers have worked to derive a single function that accurately predicts the relative permittivity of water and steam, the earliest of which being Quist and Marshall's 1965 formulation². As research in this area progressed, work was done to explore more of the temperature and pressure spectrum, refine experimental results, and propose alternate formulations in order to enhance relative permittivity prediction. Recently, Fernandez et al. compiled all of the

experimentally available data for the relative permittivity of water and steam in a single database³. Furthermore, they evaluated the methods used to experimentally derive the relative permittivity and chose a subset of the total data available that represented the most accurate values and that was recommended for use in data correlation. In 1997, Fernandez et al. proposed a new formulation based on a portion of this subset¹. Their new formulation, based on a statistical regression function, approximated the relative permittivity very well across the entire experimentally available temperature and pressure range.

In an earlier paper⁴, we proposed a formulation evolved using the genetic programming technique, to approximate the static dielectric constant across three (A, C, D) of the four thermodynamic regions (A, B, C, D) characterized by Fernandez et al.¹. This function was evolved by applying the genetic programming technique to a dataset that was recommended for data correlation by Fernandez et al. Although the proposed formulation performed comparably to Fernandez et al.'s formulation across the entire range of data values in the three thermodynamic regions (A, C, D), a formulation that can approximate the relative permittivity across the entire range of experimental values seemed both more natural and appropriate, and has been an important goal for researchers in this area. Unfortunately, the dataset for region B was incomplete (did not contain pressure values) and thus not used in our earlier formulation. In our current work, we have incorporated the available data from region B with our earlier work by evolving separate additional functions to approximate the static dielectric constant in this region. The accuracy of this more complete suite of functions is evaluated by comparing each function's output value for relative permittivity at a given temperature, pressure, and density with the output value obtained using the latest formulation of Fernandez et al., against the subset of dielectric constant values that Fernandez et al. recommended for data correlation mentioned earlier, as well as the smaller subset of values that Fernandez et al. actually used to create their formulation.

The static dielectric constant (hereon relative permittivity) of a substance, ϵ_r , is roughly defined as the ability of a substance to transmit or allow the existence of an electric field. More formally, the relative permittivity of a substance, ϵ_r , is the ratio of the static permittivity of the substance, ϵ_s , to the

static permittivity of a vacuum, ϵ_0 ¹. The relative permittivity of a substance is used for practical purposes, for example, in the design of capacitors. The behavior of the relative permittivity of water is related to its physical state or phase (as a liquid or as vapor), temperature, and pressure. Experimentally verified relative permittivity values for water in its solid phase (as ice) at temperatures as low as 190K (-83°C) exist⁵, however these data do not include corresponding pressure values for any of the measurements. Water, in its liquid or vapor (steam) state, exists within a large range of temperatures and pressures, and this range has been traditionally divided into 4 regions, A, B, C, and D. Region A is the normal liquid water state between the normal freezing and boiling points ($\sim 273\text{K}$ to $\sim 373\text{K}$) at pressures up to 1000MPa. Region B refers to water along the liquid-vapor phase boundary. For water located in this region of the thermodynamic space, in contrast to all other regions, every pair of temperature/pressure values takes on two density values (corresponding to the phase, either liquid or vapor/steam, in which the water occurs). Region C is the region with a temperature above 373.15K. At lower pressures and temperatures within region C, water is in the normal vapor (steam) state. At higher pressures and temperatures in this region, water becomes a supercritical fluid, that is, water ceases to behave as if it were in either the liquid or vapor state, but rather exhibits a combination of the thermodynamic properties attributable to both liquids and gases. Finally, region D refers to super cooled water (water that exists in the liquid state below the normal freezing point of 273.15K at the standard pressure of $\sim 0.1\text{MPa}$).

The behavior of the relative permittivity exhibits discontinuities along the liquid-vapor phase boundary (region B) and in the supercritical part of the region above the normal boiling point (region C), with very small changes in the temperature and pressure causing very large changes in density and in the value of the relative permittivity⁶. As a result, theoretical formulations for calculating the relative permittivity of water have mainly focused on a broad range of temperatures ($\sim 270\text{K}$ to $\sim 1000\text{K}$) and a narrow range of pressures ($\sim 0.1\text{MPa}$ to 100MPa)¹. Furthermore, data points along the phase boundary (region B), although numerous, have not had their pressure or density values recorded, and thus have not figured in any data-driven correlations that correct for pressure and density differences. The most

current formulation for approximating the relative permittivity across the entire range of experimental temperatures and pressures may be found in the “Fernandez database paper”³ and is also reproduced in our results section. Fernandez et al.’s 1997 formulation uses an extensive adaptive regression algorithm to create an appropriate function taking a wide variety of domain specific thermodynamic values into account. Furthermore, they analyze the first, second, and third derivatives of the change in the dielectric constant with respect to both temperature and pressure inputs in order that their function accords with theoretical considerations of how the static dielectric constant is to behave across all thermodynamic regions. Their final function uses 5 adjustable parameters and a total of 25 constants and domain specific non-adjustable parameters, and approximates well across the entire range of experimentally available values.

Genetic Programming (GP) may be seen as an abstract algorithmic implementation broadly inspired by the main principles of Darwin’s theory of evolution by means of natural selection. Roughly, Darwinian evolutionary theory involves populations of interbreeding organisms (species) competing for environmental resources over time. Species share genetic material by interbreeding, and random mutations occur to members of the species that may either hinder or further their reproductive success. As the members of a given species breed and reproduce over time, characteristics beneficial for the species’ survival propagate throughout the population, while those characteristics that are detrimental to the survival of the species do not get expressed in the population. That is, individuals with characteristics that favor their survival within the given environment tend to propagate, whereas individuals not possessing those characteristics in the environment (or those that exhibit detrimental characteristics) tend to die out.

GP applies the broad tenets of Darwinian evolutionary theory within a heuristic framework that attempts to create automatically generated computer programs that evolve to optimally solve user-defined problems⁷. GP is an extension of the evolutionary computational approach known as genetic algorithms (GA) first pioneered by John Holland^{8,9}. Within the GP framework, a population of candidate solutions, with each solution representing an executable computer program of some finite

length (an individual of a given population), evolves in response to some problem to be solved (the environmental conditions)⁷. Each GP individual/candidate computer program within the population is given a fitness value that is the output of a function (the fitness function) that determines the appropriateness or optimality of the program output (individual behavior) when given the user-defined problem (the environmental conditions). This allows each individual within the GP population to be measured against every other individual, whether the individual solves the problem (favorably responds to the environment) or not. Once all of the individuals within a population have been assigned a fitness value, certain individuals are stochastically chosen to recombine and create offspring based on their fitness values, so that individuals with higher fitness values tend to be chosen more frequently for recombination. During recombination two unique individuals are chosen to represent the parents, and may stochastically recombine to generate two offspring. Occasionally, however, (because recombination is probabilistic and does not always occur) they do not recombine and remain unchanged as offspring. After every recombination event, an offspring individual may be mutated with some small probability. The series of steps following initial population generation include parent selection, recombination, and mutation of offspring, and constitutes a generation of the GP run. At the start of every generation, newly created individuals in the population are evaluated by the fitness function and assigned a fitness value. The GP run continues in this manner (after the generation of the initial population, only fitness value assignment, parent selection, recombination, and mutation of offspring occur) until some stopping criteria (such as the creation of an individual with either some given minimum or maximum fitness value, or one that adequately solves the problem at hand) has been reached.

Each GP individual uses a tree-based data structure representation scheme, where the tree structure completely represents a given program. A tree structure resembles a company organization chart with a root node (president), subtrees (subordinate divisions under the president managed by division chiefs), and continuing down the branches until reaching leaf nodes (nodes without subordinates). Nodes for the GP program tree either come from the terminal set or the function set (both predefined by the

individual implementing the GP search). The terminal set completely defines the kinds of inputs the evolving computer programs (individuals) can use to solve the problem. The elements of the terminal set can only occur as leaf nodes within the program tree (that is, nodes that have no children). The function set defines the kinds of transformations that are permissible given any of the elements in the terminal set or any of the other elements within the function set as arguments to each of the elements within the function set. Thus, the elements of the function set may only occur as the internal nodes of a GP-generated program tree (nodes with at least one child node). These restrictions amount to the fact that the union of the function and terminal sets of a GP implementation must possess the property of closure (where closure is defined as the ability to have any composition of functions and terminals produce a syntactically correct, executable computer program¹⁰). The program trees generated using GP do not have to be standard binary trees (trees where every node is either a leaf node, or has a maximum of two child nodes), as the experimenter may define a function operator within the function set that takes more than two arguments. Initially, GP individuals are randomly generated through a stochastic tree-building process where each node in the tree is chosen to be a random member of either the function or terminal sets. Traditionally, GP candidate programs are initially generated either strictly to some maximum initial tree depth limit (where all nodes up to the maximum initial tree depth are chosen stochastically exclusively from the function set and all nodes at the maximum initial depth limit are chosen exclusively from the terminal set), or until all of the branches of the tree have either gone to the maximum initial depth or have ended in terminal nodes before the maximum initial tree depth has been reached.

The genetic programming operators of crossover and mutation, as well as the way in which individuals are ranked according to their fitness level are modified from the genetic algorithm (GA) approach to suit the GP technique⁸. Crossover occurs by selecting two nodes on different parent trees and then swapping the corresponding subtrees, that is, all of the descendants of the selected nodes (as well as the selected nodes themselves) between the two individuals.

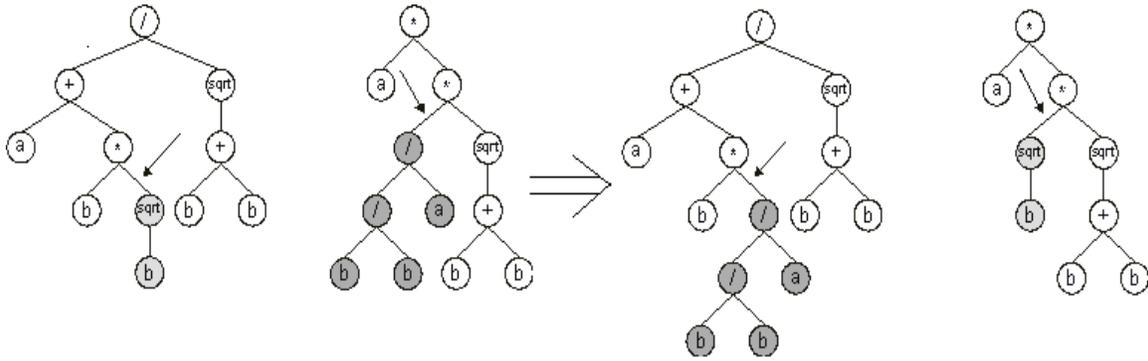


Figure 1. GP Crossover

Mutation, on the other hand, involves selecting a node at which mutation will occur, deleting all of the nodes that are descendants of the selected node, and then generating a random subtree with this node as its root.

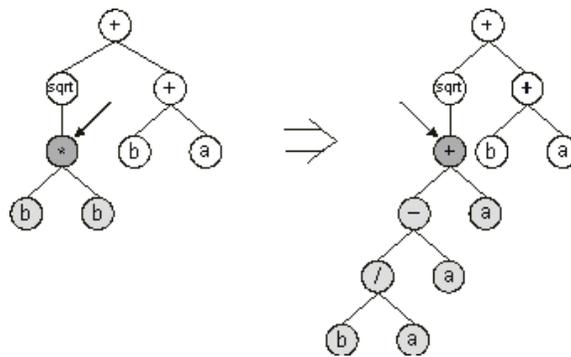


Figure 2. GP Mutation

The fitness evaluation and ranking methods in GP are slightly different from the classic GA approach (where fitness maximization is standard) in the sense that the highest ranking individual programs in GP have the lowest fitness values (in effect, a minimization problem). Thus, GP attempts to find a computer program with the globally minimal fitness value in the search space of all possible computer programs that may be created using the function and terminal sets used in the problem, to the tree depth or program length specified in the GP setup.

Ultimately, the GP approach involves determining a set of functions and terminals to be used in solving the problem, defining a fitness measure by which individual programs may be evaluated and assigned a fitness value, setting the specific parameters and operator probabilities that are involved in program tree generation (crossover and mutation probabilities, initial tree depth limit, maximum tree

length, etc.), and developing a set of rules or stopping criteria to determine when to end a specific GP run (whether after a certain number of generations have elapsed, or after an individual program with a desired fitness threshold has been found).

EXPERIMENTAL DETAILS

In our experiments, a variety of different function and terminal sets were explored in an effort to evolve a single function that could model the relative permittivity of water as a function of pressure, temperature, and density in thermodynamic regions A, C, and D of the temperature-pressure phase space. Unfortunately, no temperature *and pressure* data for region B (along the phase boundary) is currently available¹, and thus two functions approximating the dielectric constant in region B (one for vapor saturation and one for liquid saturation) taking only temperature into account were evolved separately.

The suite of functions for regions A, B, C, and D was evolved using experimentally verified input data sets taken from¹ and output results were then compared to relative permittivity values experimentally observed from the same input temperature/pressure/density values, or only input temperature values, as in the case of region B, (taken from the same data sets) using the newest formulation for dielectric constant prediction¹. These data sets were compiled from all previous experimentally available data, and were then corrected by Fernandez et al. to coincide with the most recent internationally accepted temperature scale, ITS-90. In most cases, values were provided for the temperature (in degrees Kelvin, or K), pressure (in megapascals, or MPa), and the corresponding dielectric constant. However, in some cases, temperature/density/dielectric constant values were given instead of temperature/pressure/dielectric constant values. In these circumstances, density values were converted into their corresponding pressures, and pressure values were converted to their corresponding densities using the IAPWS-95 formulation for the equation of state of water¹¹. With this completed, the final data set uniformly represented the dielectric constant at every temperature, pressure, and density value that was both accurate and experimentally available (as of December 2006).

Our functions were evolved by generating a population of possible functions (represented as trees) as with standard genetic programming implementations. Each candidate function's fitness was taken to be the sum of the absolute errors between the computed and the experimentally measured value for the relative permittivity at every input value in the corresponding data set. The combination of input values for each function (that is, what combination of the three possible adjustable inputs was to be used) was determined by the GP module. The population of possible functions was then evolved with a variety of crossover/mutation probabilities and function sets. The data set of experimentally obtained relative permittivity values used to create the function consisted of 771 data points, which represent the complete dataset that Fernandez et al. recommend for data correlations for regions A, C, and D (644 data points) and region B (127 data points)³. The function with the lowest sum of absolute errors across the data points that was found after all runs had been completed was chosen as the final function for each part of the formulation.

During any given GP run, all function and terminal sets used during function evolution always included addition, subtraction, multiplication, and division as function operators, and temperature, T_k , pressure, p , and density, ρ , as terminal values. In cases where a generated function divided a value by zero, the zero-generating term was replaced by 0.00001. All runs used a population of 100 random floating-point constants in the range between 0 and 1, which were generated at runtime. These constants would play the role of additional terminal values for the genetic program to use during function creation. Other function operators ($\sin()$, $\cos()$, $\ln()$, \log_{10} , \log_2 , and x^y) and terminal operators (Avogadro's number, N_A ; permittivity of free space, ϵ_0 ; elementary charge, e ; Boltzmann's constant, k ; molar mass of water, M_w ; mean molecular polarizability of water, α ; and the dipole moment of water, μ) were also used in various GP runs. The aforementioned terminal operators are reproduced from³ in table 1. The function length of any individual solution (a tree representing a given candidate function) never exceeded 100 functional units (where a functional unit is taken to be a single operator from the function set or a terminal value from the terminal set). The large size of the function

and terminal sets causes the size of the search space (representing all of the possible unique programs of length 100 or less that can be generated from the function and terminal sets) to be enormous (more than a googol possible syntactically correct possible functions). As a result, each GP run was done on a population of two million individuals that were evolved for 200 generations. This was done to ensure that the GP implementation would sample as much of the search space as possible in its effort to find a suitable function within a reasonable time. A range of crossover probabilities (between .5 and 1.0, in increments of .05) and mutation probabilities (between 0 and .5, in increments of .05) were explored for all combinations of function and terminal sets. Each combination of unique parameter settings was implemented in 10 GP runs, after which the function with the lowest total absolute error was chosen.

RESULTS

The best functions that were evolved were found during runs that used multiplication, division, subtraction, and addition as operators in the function set and temperature, pressure, and density as terminal operators (with the 100 additional random ephemeral constants described earlier). The best function runs used a crossover probability of 0.9 and a mutation probability of 0.05. The final evolved functions along with Fernandez, et. al.'s formulation, follow:

$$\varepsilon_{A,C,D} = \frac{\frac{\rho^2 + \rho^3}{-.02036T_K\rho + .0864\rho^2 + .1194T_K p} + \frac{(-6.75862p^2 + .313T_K p - T_K^2)(1 + \rho)}{T_K}}{55.474T_K p + 55.55p^2 - .076T_K p\rho + .016p + .016p\rho + \rho + \rho^2} + \frac{\rho^2}{.03264T_K(T_K + \rho)} - \frac{\frac{T_K^2}{\rho^2} + 2.617p - 1.617\rho + 1.617T_K}{T_K + .0486 - p + \rho}$$

Figure 3. GP-Evolved Equation, regions A, C, and D

$$\mathcal{E}_{B,liquid} = \frac{0.0209}{88000 + \frac{T_K^3 - 0.2864T_K^2}{T_K^2 - 0.8234}} + 0.1553 - \frac{87968.427}{T_K^2 + 0.97T_K + 0.196} * \frac{-0.31093*(T_K + 0.567)*(T_K + 1.6)}{T_K} - \frac{0.2038}{T_K^2 + 0.0896T_K}$$

Figure 4. GP-Evolved Equation, region B, liquid

$$\mathcal{E}_{B,vapor} = \frac{T_K - 0.33439}{-0.0015136T_K^2 + 2T_K - 1.08451} - \frac{0.108579T_K}{-0.003027T_K^2 + 1.8963794T_K + 1.4214339} + \frac{0.1085797T_K}{-0.003027T_K^2 + 2T_K + 1.05097} + 0.067589$$

Figure 5. GP-Evolved Equation, region B, vapor

$$\varepsilon_r = \frac{1 + 5A + 5B + \sqrt{9 + 2A + 18B + A^2 + 10AB + 9B^2}}{4 - 4B}$$

where A and B are given by

$$A = \frac{N_A \mu^2}{\varepsilon_0 k} \frac{\rho g}{T_k} \quad B = \frac{N_A \alpha}{3\varepsilon_0} \rho$$

and where g is given by

$$g = 1 + \sum_{k=1}^{11} N_k \left(\frac{\rho}{\rho_c}\right)^{i_k} \left(\frac{T_c}{T}\right)^{j_k} + N_{12} \left(\frac{\rho}{\rho_c}\right) \left(\frac{T}{228K} - 1\right)^{-q}$$

with $\rho_c = \frac{322}{M_w}$ and $T_c = 647.096K$ and values for N_k, i_k, j_k , and q given in table 2.

Figure 6. Fernandez' formulation¹, reproduced from Fernandez et al.

The results of applying the GP-evolved functions and Fernandez et al.'s formulation to the total data set are found in tables 3 through 6. The evolved functions shown above are significantly smaller than the formulation developed by Fernandez et al. (31 terms for the regions A, C, D function, 24 terms for the liquid saturation function, and 17 terms for the vapor saturation function versus 112 terms for Fernandez et al.'s formulation) and uses only three adjustable parameters (temperature, pressure, and density), zero non-adjustable domain specific parameters, and only fifteen of the one hundred possible random ephemeral constants that were available during function evolution. No domain-specific knowledge (aside from the data sets themselves) was applied to the formulation of the suite of functions.

As can be seen from tables 3 through 5, the evolved function performed comparably to Fernandez et al.'s formulation in all collected statistical categories except the minimum absolute difference, where each function had at least one data point where very marginal absolute error (<0.01) existed. However, it must be mentioned that our proposed formulation and Fernandez et al.'s formulation differ in the way each was generated. Fernandez et al. did not use the entire set of data points that they recommended for data correlation, but rather a significantly reduced subset of this total dataset (127 data points out of the total 771 data points for all four regions). Table 6 compares the accuracy of our formulation to that of Fernandez et al. on this reduced dataset. As should be expected, our formulation does not perform as well as Fernandez et al.'s on these selected data points, but does have a smaller maximum absolute difference across the data set. This dataset does not cover the entire thermodynamic space, and thus, Fernandez et al. used a variety of theoretical considerations to buttress the sparseness of their dataset. In the case of region B, Fernandez et al. also provide pressure and density values for the data points that they used in constructing their formulation, even though no experimental pressures and densities for this region exist. Furthermore, Fernandez et al. weighted each data point in their dataset differently, based on certain theoretical and experimental factors. Although it was our goal to create a formulation following steps as similar to Fernandez et al.'s as possible, we could not follow this aspect of their experimental methodology. This approach entailed creating copies of each data point in proportion to its weight and then reinserting those copies back into the data set or multiplying the error on each data point in proportion to its weight. This seems tractable until one realizes that the weights of some data points are very small (<0.01 , where the sum of the weights of all 127 data points is 100). As a result, certain data points with much larger weight values (weights between 1 and 3) would skew the evolutionary search to find functions that approximate those points well, but approximate the low-weight data points very poorly. Furthermore, this would come at the expense of not covering the entire thermodynamic space (even when other experimental points that do cover the thermodynamic space exist). Thus, we decided to leave the weights of all data points equal, and to use all of the experimental data that was available and recommended for correlation.

CONCLUSIONS

A suite of functions that approximate the relative permittivity of water and steam across the entire experimentally verified range of temperatures and pressures has been developed. These functions were evolved using the GP technique with a specific function and terminal set, and their accuracy has been compared to that achieved by Fernandez et al.'s most recent formulation. This approach uses no theoretical domain-specific knowledge to obtain a useable function. The evolved functions approximate the relative permittivity of water and steam extremely well, comparing favorably with Fernandez et al.'s formulation across the entire experimentally available temperature and pressure range, while being simpler computationally. Further refinements to create more accurate approximations of the relative permittivity of water and steam will include creating a single evolved function that can be used across all thermodynamically distinct temperature and pressure regions, including regions where water is in the solid phase, or where a phase boundary exists. This can be done when experimental values for the temperature, pressure, and relative permittivity in these regions (especially region B) are obtained. A refined fitness function that takes more than the absolute difference between expected and calculated values may also prove useful in creating a new, more accurate formulation. Introducing a penalty for very large and difficult to read functions may also help in finding a function that is both compact and generalizes well across the entire thermodynamic space. However, significant improvements to the evolution of an appropriate function will most surely come from an increase in experimentally verifiable values for the relative permittivity, and thus any new accurate data that may be found should be used to refine the current formulation.

ACKNOWLEDGEMENT: The authors wish to express their utmost appreciation and thanks to Dr. W. L. Marshall (ORNL, retired) for all his help and encouragement during this research project.

TABLES

Parameter	Value
Permittivity of free space, ϵ_0	$[4 * 10^{-7} \pi (299792458)^2]^{-1} C^2 J^{-1} m^{-1}$
Elementary charge, e	$1.60217733 * 10^{-19} C$
Boltzmann's constant, k	$1.380658 * 10^{-23} JK^{-1}$
Avogadro's number, N_A	$6.0221367 * 10^{23} mol^{-1}$
Molar mass of water, M_w	$0.018015268 kg * mol^{-1}$
Mean molecular polarizability of water, α	$1.636 * 10^{-40} C^2 J^{-1} m^{-2}$
Dipole moment of water, μ	$6.138 * 10^{-30} Cm$

Table 1. Constants used in the relative permittivity formulation, reproduced from³.

k	N_k	i_k	j_k
1	0.978224486826	1	0.25
2	-0.957771379375	1	1
3	0.237511794148	1	2.5
4	0.714692244396	2	1.5
5	-0.298217036956	3	1.5
6	-0.108863472196	3	2.5
7	$0.949327488264 * 10^{-1}$	4	2
8	$-0.980469816509 * 10^{-2}$	5	2
9	$0.165167634970 * 10^{-4}$	6	5
10	$0.937359795772 * 10^{-4}$	7	0.5
11	$-0.123179218720 * 10^{-9}$	10	10
12	$0.196096504426 * 10^{-2}$		$q=1.2$

Table 2. Coefficients N_k , and exponents i_k , j_k , and q of the equation for g, reproduced from³.

REGIONS A,C,D	Evolved GP result	Fernandez
Sum Absolute Difference	103.12	95.20
Mean Absolute Difference	0.16	0.15
Standard Deviation Absolute Difference	0.25	0.22
Sum Squared Difference	55.80	44.30
Mean Squared Difference	0.09	0.07
Standard Deviation Squared Difference	0.54	0.54
Minimum Absolute Difference	0.00	0.00
Maximum Absolute Difference	3.55	3.60
# Data Points Absolute Difference formulation < Absolute Difference Fernandez	330	
% Total Data Points better than Fernandez	50.93%	
Total Data Points	644	

Table 3. Results and numeric comparison, regions A, C, D

REGION B, liquid	Evolved GP result	Fernandez
Sum Absolute Difference	27.11	37.49
Mean Absolute Difference	0.22	0.30
Standard Deviation Absolute Difference	0.18	0.33
Sum Squared Difference	9.74	24.64
Mean Squared Difference	0.08	0.20
Standard Deviation Squared Difference	0.13	0.65
Minimum Absolute Difference	0.00	0.00
Maximum Absolute Difference	0.90	2.56
# Data Points Absolute Difference formulation < Absolute Difference Fernandez	71	
% Total Data Points better than Fernandez	56.35%	
Total Data Points	126	

Table 4. Results and numeric comparison, region B, liquid

REGION B, vapor	Evolved GP result	Fernandez
Sum Absolute Difference	0.273	0.335
Mean Absolute Difference	0.007	0.009
Standard Deviation Absolute Difference	0.016	0.017
Sum Squared Difference	0.011	0.013
Mean Squared Difference	0.0003	0.0004
Standard Deviation Squared Difference	0.001	0.001
Minimum Absolute Difference	0.00	0.00
Maximum Absolute Difference	0.075	0.074
# Data Points Absolute Difference formulation < Absolute Difference Fernandez	22	
% Total Data Points better than Fernandez	59.46%	
Total Data Points	37	

Table 5. Results and numeric comparison, region B, vapor

REGIONS A,B,C,D	Evolved GP result	Fernandez
Sum Absolute Difference	26.25	16.18
Mean Absolute Difference	0.21	0.13
Standard Deviation Absolute Difference	0.37	0.34
Sum Squared Difference	22.70	16.38
Mean Squared Difference	0.18	0.13
Standard Deviation Squared Difference	1.13	1.15
Minimum Absolute Difference	0.00	0.00
Maximum Absolute Difference	3.55	3.60
# Data Points Absolute Difference formulation < Absolute Difference Fernandez	40	
% Total Data Points better than Fernandez	31.50%	
Total Data Points	127	

Table 6. Results and numeric comparison (points used for correlation), ALL REGIONS.

REFERENCES

- (1) Fernandez, D.P.; Mulev, Y.; Goodwin, A.R.H.; Levelt-Sengers, J.M.H. *J. Phy. Chem. Ref. Data*, **1995**, *24*, 33.
- (2) Quist, A.S.; Marshall, W.L. *J. Phy. Chem.* **1965**, *9*, 3165.
- (3) Fernandez, D.P.; Goodwin, A.R.H.; Lemmon, E.W.; Levelt-Sengers, J.M.H.; Williams, R.C.. *J. Phy. Chem. Ref. Data* **1997**, *26*, 1125.
- (4) Fogelson, S.; Potter, W. 2007. *Proc. of Int. Conf. on A.I. and Patt. Rec., AIPR'07*, Orlando, FL (to appear).
- (5) Matsuoka, T.; Fujita, S.; Mae, S. *J. App. Phys.*, **1996**, *80*, 5884.
- (6) Harvey, A. NIST. Personal Communication.
- (7) Koza, J.R. 1992. *Genetic Programming*; MIT Press: Cambridge, MA, 1992.
- (8) Holland, J. *Adaptation in Natural and Artificial Systems: 2nd Edition*; MIT Press: Cambridge, MA, 1992.
- (9) Forrest, S. *Science* **1993**, *261*, 872.
- (10) Ghanea-Hercock, R. *Applied Evolutionary Algorithms in Java*; Springer: New York, NY, 2003.
- (11) Wagner, W; Pruss, A. 2002. *J. Phy. Chem Ref. Data* **2002**, *31*, 387.