Using Gene-expression Programming (GEP) to describe Saturated Adiabats

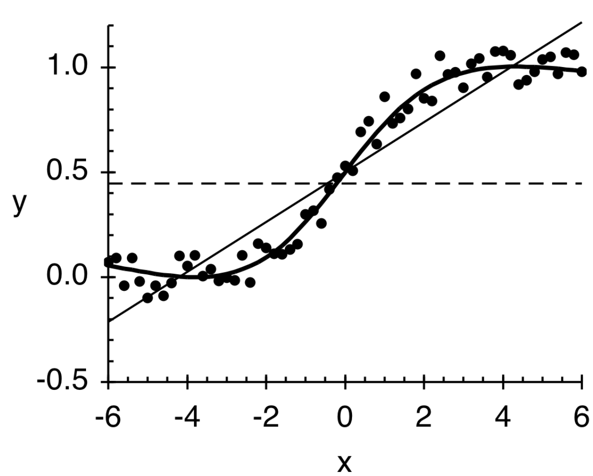
Roland Stull
UBC

Capabilities of GEP

- Finding a function to fit noisy data.
  Example: Fig. 2 from Atoossa & Stull:

$$y = \frac{(x+3.7856)/3.7856}{2+0.466x\left(\frac{x-3.7856}{x+3.7856}\right)}$$

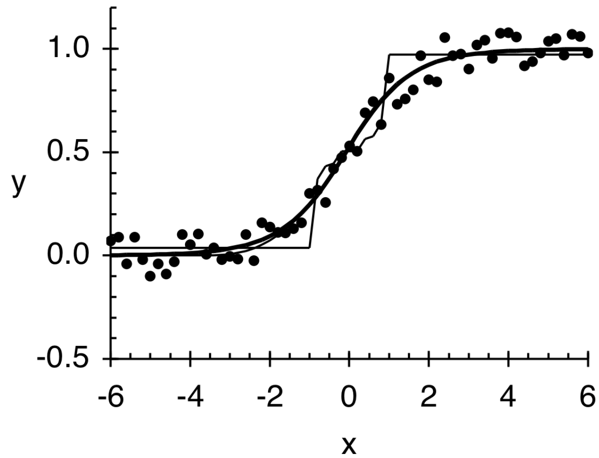
d)



Capabilities of GEP

- Inexorable approach to a good fit.

Example: Fig. 3 from Atoossa & Stull:

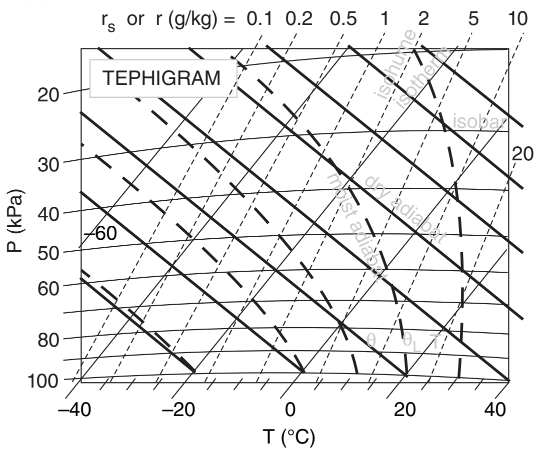


allowable functions is limited to (+, – , L) where L is the logistic function $L(x) = \{ 1 + \exp(-x) \}^{-1}$, then GEP evolves to a fittest individual (thick line) of the form $y = L\{-0.38745 + x + L(x)\}$, with fitness $r^2 = 0.978$. (b) If GEP functions are limited to (+, – , Power), then the fittest individual (thin dashed line, mostly hidden behind the thick line) is $y = 0.762055^{\left[2.388183^{\{(0.881012^x)-x\}}\right]}$, with $r^2 = 0.978$. (c) If the GEP functions are limited to (+, –, mod) where "mod" is the floating-point remainder (as defined in the Fortran 95/2003 standard, not as defined in Microsoft Excel), then the fittest individual (thin solid line) is $y = 0.5057 + \mathrm{mod}\left[\mathrm{mod}\left\{\mathrm{mod}(x,(-0.468079-x)),(-0.468079+x)\right\},x\right]$, with $r^2 = 0.952$. These last two

Opportunity

- Identify simple, but long-standing problems/ desires in meteorology that could be solved using GEP.
- My first experiment:
  saturated adiabats on a Tephigram.

Define the Problem



where $T_2 = T_1 + (\Delta T/\Delta P)\cdot(P_2 - P_1)$

$$\frac{\Delta T}{\Delta P} = \frac{\left[\left(\Re_d / C_p\right)\cdot T + \left(L_v / C_p\right)\cdot r_s\right]}{P\cdot\left(1 + \frac{L_v^{\,2}\cdot r_s\cdot\varepsilon}{C_p\cdot\Re_d\cdot T^2}\right)}$$

where $r_s = \varepsilon\cdot e_s/(P - e_s)$, and $e_s = \text{fnt}(T)$

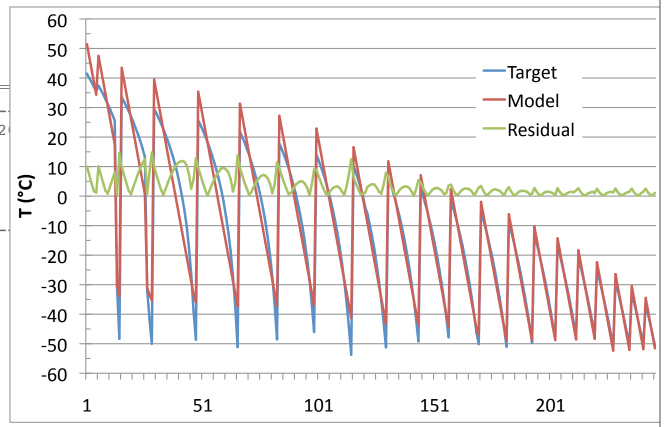
- For most processes, if we are given the curve and the pressure of interest (P), we can directly calculate T. For example, for any dry adiabat θ we can directly calculate T(P, θ) from an eq.
- But for the saturated adiabat θ_w, we need to iterate an eq. from P = 100 kPa where θ_w = T is defined up to the P of interest, in order to find T(P, θ_w).

Goal

- Wouldn't it be nice if GEP could give us a deterministic expression for T(P, θ_w).
- So, I will pick off values of T, P, θ_w from a Tephigram, and feed them into GEP, and ask GEP to find a function for T(P, θ_w).

	A	B	C
1	ThetaWC	PkPa	TwetC
2	24	105	25.7
3	24	100	24
4	24	95	22.3
5	24	90	20.3
6	24	85	18.4
7	24	80	16.1
8	24	75	13.8
9	24	70	11.2
10	24	65	8.4
11	24	60	5.3
12	24	55	2
13	24	50	-2.1
14	24	45	-6.9
15	24	40	-12.2
16	24	35	-18.7
17	24	30	-27.2
18	24	25	-37.7
19	24	20	-51.1
20	20	105	21.8
21	20	100	20
22	20	95	18.2
23	20	90	16
24	20	85	13.9
25	20	80	11.4
26	20	75	8.9
27	20	70	6.1
28	20	65	3
29	20	60	-0.4
30	20	55	-4.3
31	20	50	-8.8
32	20	45	-14
33	20	40	-20.1
34	20	35	-27.6
35	20	30	-37
36	20	25	-48.5

How well did GEP work?




```
!-----------------------------------------------------
! Code generated by GeneXproTools 4.0 on 03/03/2
! Training Samples:  246
! Fitness  Function: RRSE
! Training Fitness:  837.72824985952
! Training R-square: 0.963223027172055
!-----------------------------------------------------

    real function gepModel(d)

        real, intent(in) :: d(0:1)

        real, parameter :: G1C0 = 8.82544
        real, parameter :: G1C1 = -3.335327
        real, parameter :: G2C0 = 0.85495
        real, parameter :: G2C1 = 9.354156
        real, parameter :: G3C0 = 5.393677
        real, parameter :: G3C1 = 9.864624

        real :: varTemp
        varTemp = 0.0

        varTemp = (d(0)-(G1C0**2))
        varTemp = varTemp + (((G2C0*d(0))-d(0))+atan((d(1)-G2C1)))
        varTemp = varTemp + (d(1)-(log((G3C0**2))-atan((d(1)/G3C0))))

        gepModel = varTemp

    end function gepModel
```

Mediocre

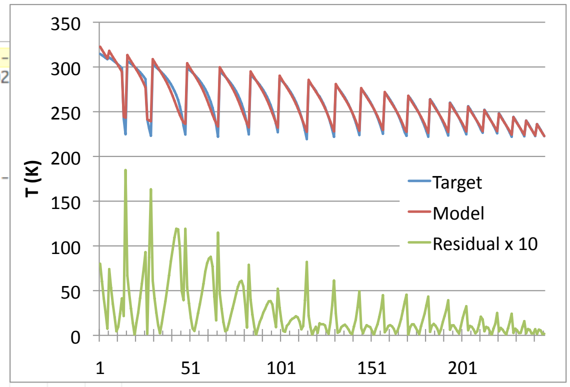
Peculiarities of GEP

- Some functions have sharp changes across x = 0.   e.g. abs(x)
- Some functions are undefined for negative numbers.   e.g. ln(x)
- T = 0°C is not a special number.
- Solution: Perhaps can change °C to Kelvin to avoid this problem.

ThetaWK	PkPa	TwetK
297.15	105	298.85
297.15	100	297.15
297.15	95	295.45
297.15	90	293.45
297.15	85	291.55
297.15	80	289.25
297.15	75	286.95
297.15	70	284.35
297.15	65	281.55
297.15	60	278.45
297.15	55	275.15
297.15	50	271.05
297.15	45	266.25
297.15	40	260.95
297.15	35	254.45
297.15	30	245.95
297.15	25	235.45
297.15	20	222.05
293.15	105	294.95
293.15	100	293.15
293.15	95	291.35
293.15	90	289.15
293.15	85	287.05
293.15	80	284.55
293.15	75	282.05
293.15	70	279.25
293.15	65	276.15
293.15	60	272.75
293.15	55	268.85
293.15	50	264.35
293.15	45	259.15
293.15	40	253.05
293.15	35	245.55
293.15	30	236.15
293.15	25	224.65

How well did GEP work?

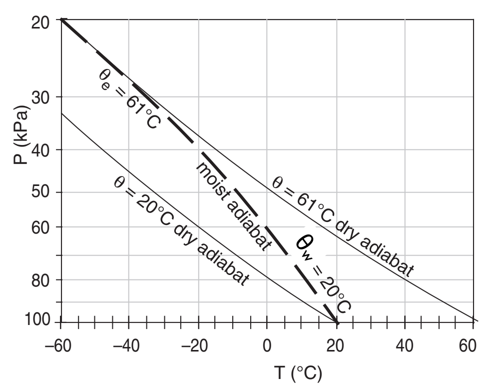
```
!-----------------------------------------------------------
! Code generated by GeneXproTools 4.0 on 03/03/2009 8:02
! Training Samples:  246
! Fitness  Function: RRSE
! Training Fitness:  866.465722781635
! Training R-square: 0.976331602056821
!-----------------------------------------------------------

    real function gepModel(d)

        real, intent(in) :: d(0:1)

        real, parameter :: G1C0 = -7.840851
        real, parameter :: G1C1 = -9.934205
        real, parameter :: G2C0 = 3.905793
        real, parameter :: G2C1 = 4.713165
        real, parameter :: G3C0 = -2.181244
        real, parameter :: G3C1 = -6.156738

        real :: varTemp
        varTemp = 0.0

        varTemp = ((G1C0+sin(sqrt(d(0))))*((d(0)-G1C1)**(1.0/3.0)))
        varTemp = varTemp + (d(1)-(cos((d(1)**(1.0/3.0)))*(G2C0**3)))
        varTemp = varTemp + (d(0)-sqrt((sin((G3C1*d(1)))-G3C0)))

        gepModel = varTemp

    end function gepModel
```



Mediocre

Think First. Then Do.

- All the saturated adiabats have similar behavior:
  - at bottom, $\theta = \theta_{dry}$.
  - at top, θ is asymptotic to θ_e.

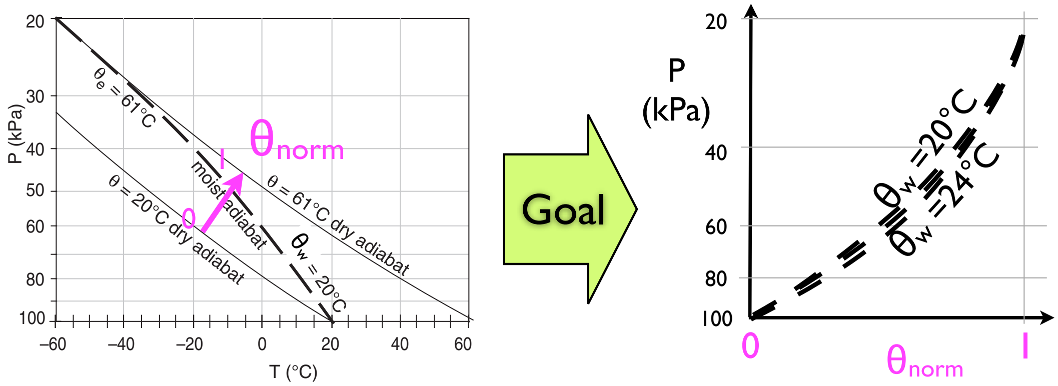


- Also, the there are deterministic eqs for the dry adiabats shown in this fig, and for the relationship between θ_e and θ_w:

$$\theta_e = \theta_w + (Lv/Cp) \cdot r_{so}$$

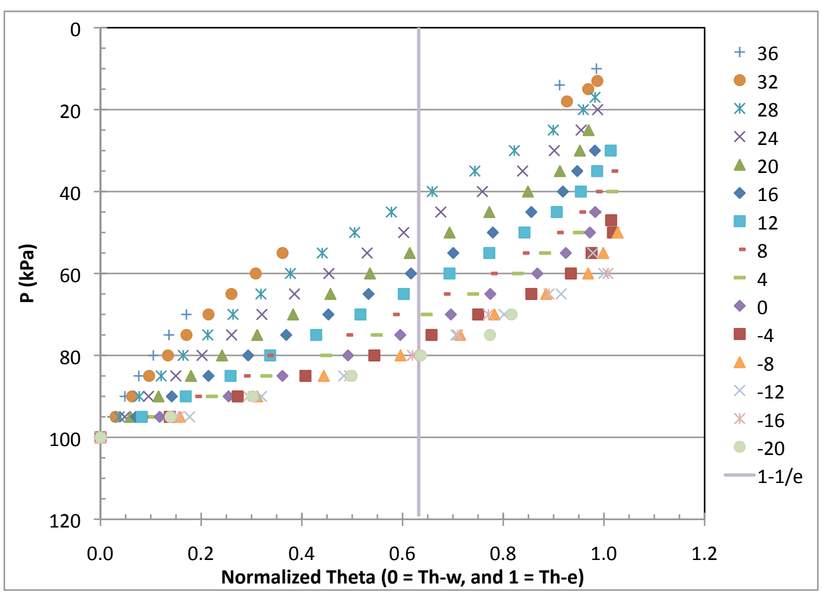
where r_{so} is the saturated mixing ratio at P = 100 kPa and T = θ_w.

Think First.   Then Do.



- So perhaps I can normalize the curves by
  $\theta_{norm.} = (\theta - \theta_{dry}) / (\theta_e - \theta_{dry})$.
  Thus: along any one sat. adiabat, $\theta_{norm.}$ = 0 at the bottom, and approaches $\theta_{norm.}$ = 1 at top.
- If I am lucky, perhaps all the normalized curves will collapse into a single shape. (i.e., similarity theory).

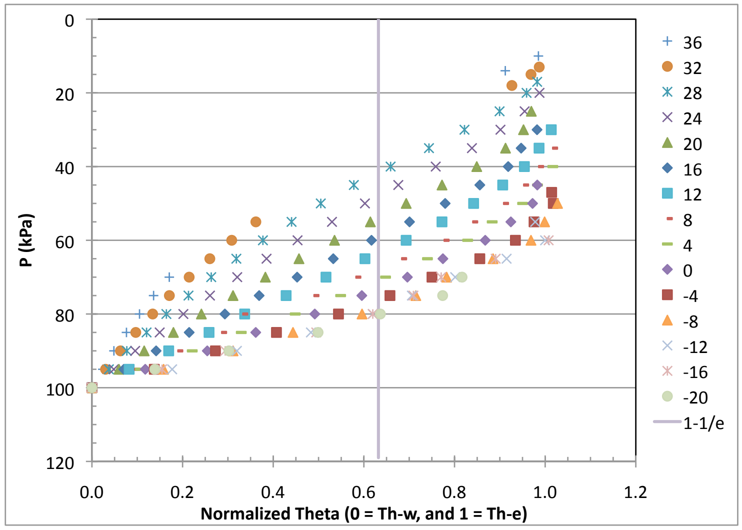
I wasn't lucky.



However, I did succeed in data reduction. All these curve start at the same point.

Think First. Then Do.

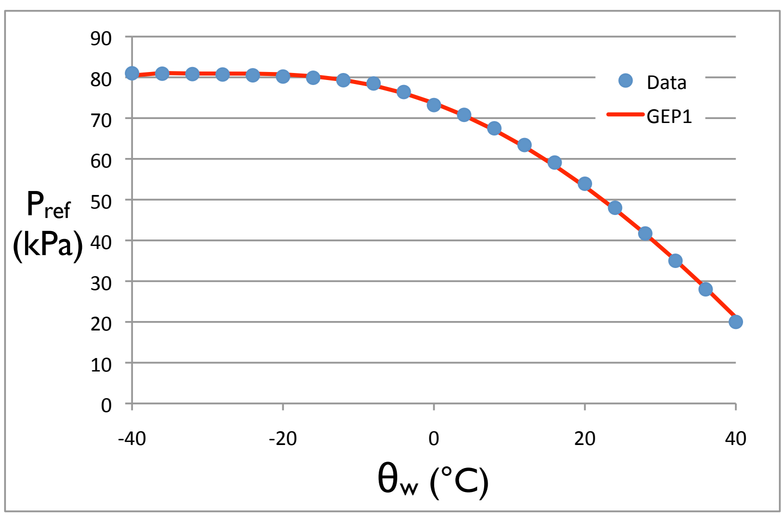
- Perhaps I can similarly normalize the pressure P.
- After some thought, I decided to normalize by the pressure P_{ref} where θ_{norm} = (1 - 1/e) = 0.632



- Namely, define $Y = (100 - P) / (100 - P_{ref})$

Is there a relationship between P_{ref} and θ_w ?

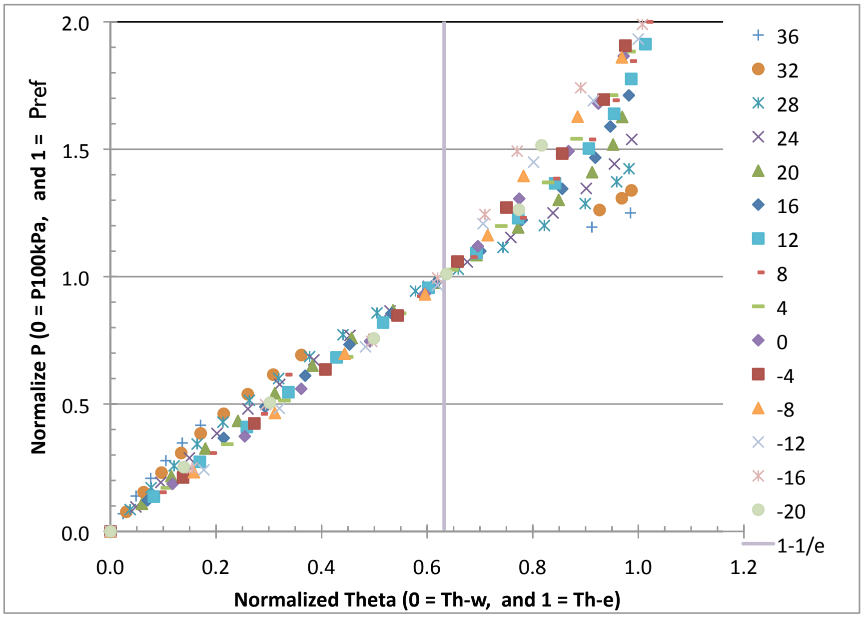
- Yes, and GEP was able to easily fit it.



- Good. I can use: $Y = (100 - P) / (100 - P_{ref})$
- If I am lucky, perhaps all the normalized curves Y vs. θ_{norm} will collapse into a single shape. (i.e., similarity theory).

I was almost lucky.

Y

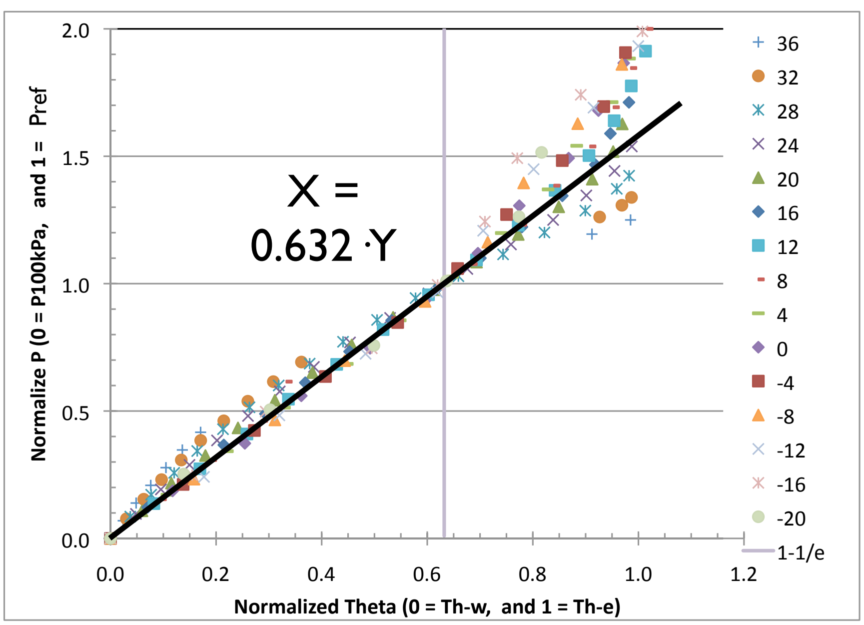


θnorm

The curves almost collapse onto a straight line.

Best fit straight line

Y



θnorm

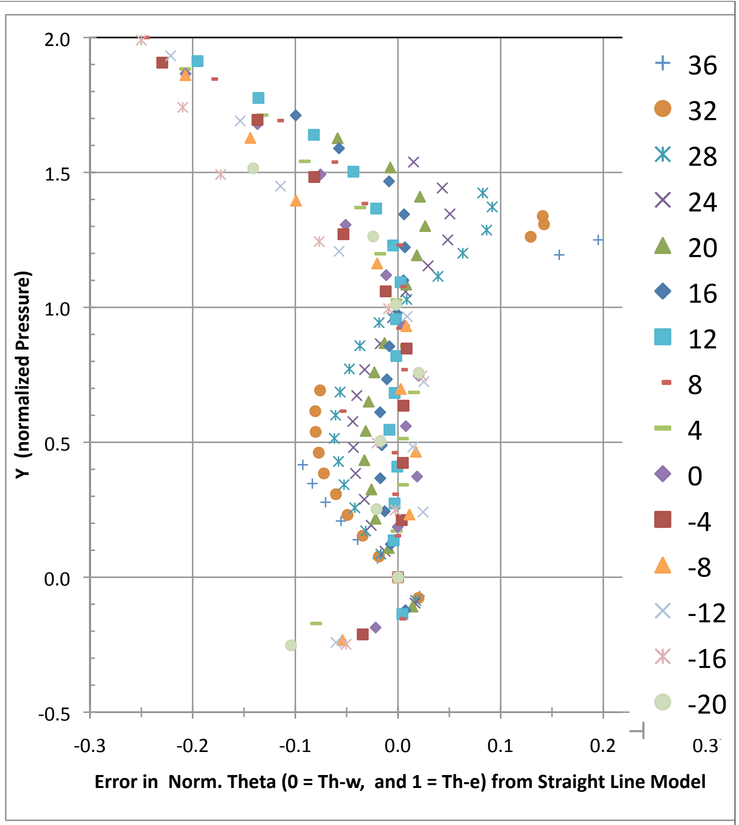
Perhaps this is good enough. It yields the following steps to get $T(P, \theta_w)$:

1) Get $P_{ref}(\theta_w)$ using the GEP algorithm.
2) Calculate $Y = (100 - P) / (100 - P_{ref})$.
3) Calculate $\theta_{norm} = 0.632 \cdot Y$.
4) Find $\theta_w(K) = \theta_w(C) + 273$.
5) Find $e_s(\theta_w)$ from Clausius-Clapeyron eq.
6) Find $r_s = 0.622\, e_s / (100 - e_s)$.
7) Find $\theta_e(K) = \theta_w(K) \cdot \exp(2600\, r_s/\theta_w(K))$.
8) Find $\theta(K) = \theta_w(K) + \theta_{norm} \cdot [\theta_e(K) - \theta_w(K)]$
9) Find $T(K) = \theta(K) \cdot [P/100]^{0.28571}$
10) Find $T(°C) = T(K) - 273$.

But how much error remains? Answer: subtract straight line from previous graph.

Remaining Error

Error is still fairly large.



Conclusions

- It was an interesting exercise.
- I learned some more about GEP.
- I learned a lot more about thermodynamics.
- I reinforced my recommendation to think first, then do.
- Will the result be use useful to meteorologists as I had hoped? Probably not.

The End. Any Questions?