

Temperature dependent adjustment of V_c and V_o

Bernacchi temperature functions

Currently the e-Photosynthesis model sets $PrV111 = PsV1 * 0.24$, according to the known value for tobacco at 25°C (Whitney et al., 1999).

However, this ratio varies across different crop species and temperatures.

Seeing as there is no existing temperature functions for rice, we will use the equations by Bernacchi (2001) and values from Makino (1988) to compare the temperature responses of the parameters.

Specify constants:

```
T25 = 25 + 273.15;           % Reference temperature in K
Tp = 28.9310407291759 + 273.15; % Average measurement temperature in K
R = 8.314/1000;              % Ideal gas constant in kJ K-1 mol-1
c_Vc = 26.35;                % Scaling constant, Vcmax (N.
tabacum L. cv. W38, Bernacchi et al 2001)
dHa_Vc = 65.33;              % Activation energy, Vcmax, kJ mol -1 (N.
tabacum L. cv. W38, Bernacchi et al 2001)
c_Vo = 22.98;                % Scaling constant, Vomax (N.
tabacum L. cv. W38, Bernacchi et al 2001)
dHa_Vo = 60.11;              % Activation energy, Vomax, kJ mol -1 (N.
tabacum L. cv. W38, Bernacchi et al 2001)
```

Use the Arrhenius equation $Parameter = \exp\left(c - \frac{\Delta H_a}{RT_K}\right)$ to calculate $\frac{V_o}{V_c}$ ratio at 25°C for N.tabacum (c and dHa from Bernacchi, et al., 2001):

```
Bernacchi_Vc_25 = exp(c_Vc - dHa_Vc / (R * T25));
Bernacchi_Vo_25 = exp(c_Vo - dHa_Vo / (R * T25));
Bernacchi_PrPs_ratio_25 = Bernacchi_Vo_25/Bernacchi_Vc_25;
```

Specify a range of temperatures for calculating Vc and Vo ratios:

```
Temp_Range(:,1) = (10:1:40)'; % Range of
temperatures in C
Vo_Vc_matrix = deal(zeros(length(Temp_Range),5)); % Output matrix
```

Loop the Arrhenius equation to calculate values over the range of temperatures:

```
for i = 1:length(Temp_Range)
Vo_Vc_matrix(i,1) = Temp_Range(i,1); %
Range of temperatures in C
Vo_Vc_matrix(i,2) = Temp_Range(i,1)+273.15; %
Range of temperatures in K
```

```

Vo_Vc_matrix(i,3) = exp(c_Vc - dHa_Vc / (R * Vo_Vc_matrix(i,2)));           % Vc
values
Vo_Vc_matrix(i,4) = exp(c_Vo - dHa_Vo / (R * Vo_Vc_matrix(i,2)));           % Vo
values
Vo_Vc_matrix(i,5) = Vo_Vc_matrix(i,4)/Vo_Vc_matrix(i,3);                     %
Vo/Vc ratios
end

```

Plot Vo/Vc data:

```

figure
Vo_Vc_plot=plot(Temp_Range(:,1),Vo_Vc_matrix(:,5))

```

```

Vo_Vc_plot =
  Line with properties:
      Color: [0 0.4470 0.7410]
  LineStyle: '-'
  LineWidth: 0.5000
    Marker: 'none'
  MarkerSize: 6
  MarkerFaceColor: 'none'
      XData: [10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40]
      YData: [0.3158 0.3134 0.3109 0.3086 0.3062 0.3039 0.3016 0.2994 0.2971 0.2950 0.2928 0.2907 0.2886 0.2865 0.2843 0.2821 0.2799 0.2777 0.2755 0.2733 0.2711 0.2689 0.2667 0.2645 0.2623 0.2601 0.2579 0.2557 0.2535 0.2513 0.2491 0.2469 0.2447 0.2425 0.2403 0.2381 0.2359 0.2337 0.2315 0.2293 0.2271 0.2249 0.2227 0.2205 0.2183 0.2161 0.2139 0.2117 0.2095 0.2073 0.2051 0.2029 0.2007 0.1985 0.1963 0.1941 0.1919 0.1897 0.1875 0.1853 0.1831 0.1809 0.1787 0.1765 0.1743 0.1721 0.1699 0.1677 0.1655 0.1633 0.1611 0.1589 0.1567 0.1545 0.1523 0.1501 0.1479 0.1457 0.1435 0.1413 0.1391 0.1369 0.1347 0.1325 0.1303 0.1281 0.1259 0.1237 0.1215 0.1193 0.1171 0.1149 0.1127 0.1105 0.1083 0.1061 0.1039 0.1017 0.0995 0.0973 0.0951 0.0929 0.0907 0.0885 0.0863 0.0841 0.0819 0.0797 0.0775 0.0753 0.0731 0.0709 0.0687 0.0665 0.0643 0.0621 0.0599 0.0577 0.0555 0.0533 0.0511 0.0489 0.0467 0.0445 0.0423 0.0401 0.0379 0.0357 0.0335 0.0313 0.0291 0.0269 0.0247 0.0225 0.0203 0.0181 0.0159 0.0137 0.0115 0.0093 0.0071 0.0049 0.0027 0.0005 0.0000]

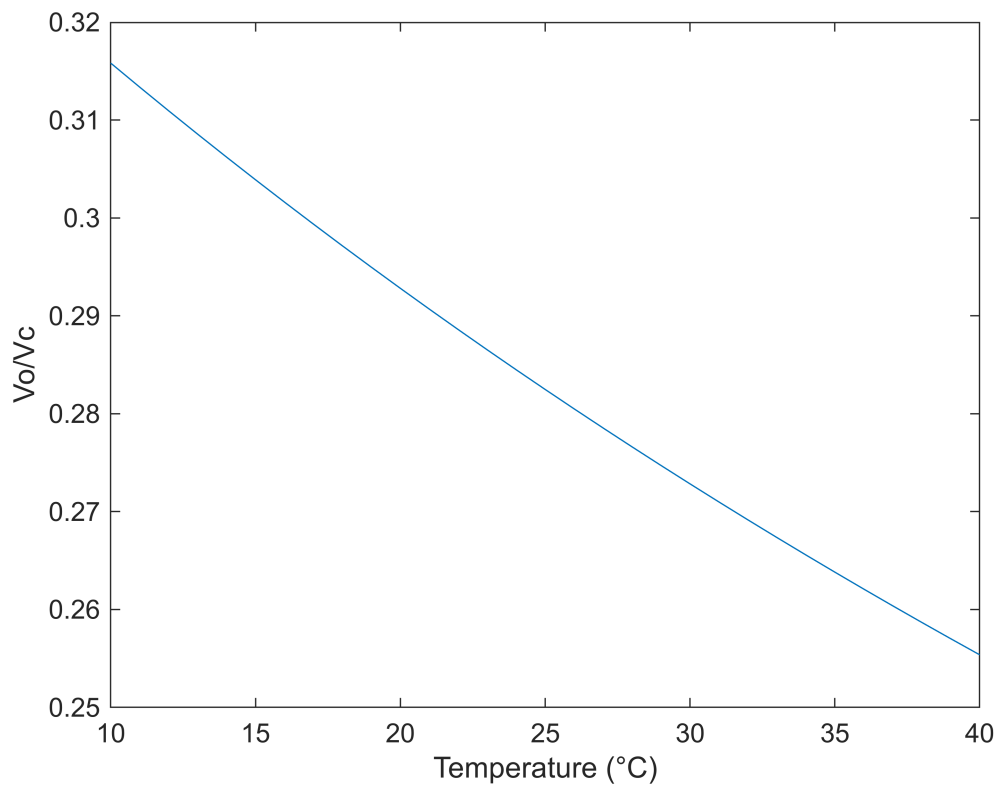
```

Show all properties

```

xlabel('Temperature (°C)')
ylabel('Vo/Vc')
set(gcf, 'PaperOrientation', 'landscape');
print(gcf,fullfile('Outputs/rice_params/graphs',"Temp_vs_Vo_Vc"),'-djpeg');

```



Plot Vc and Vo for comparison:

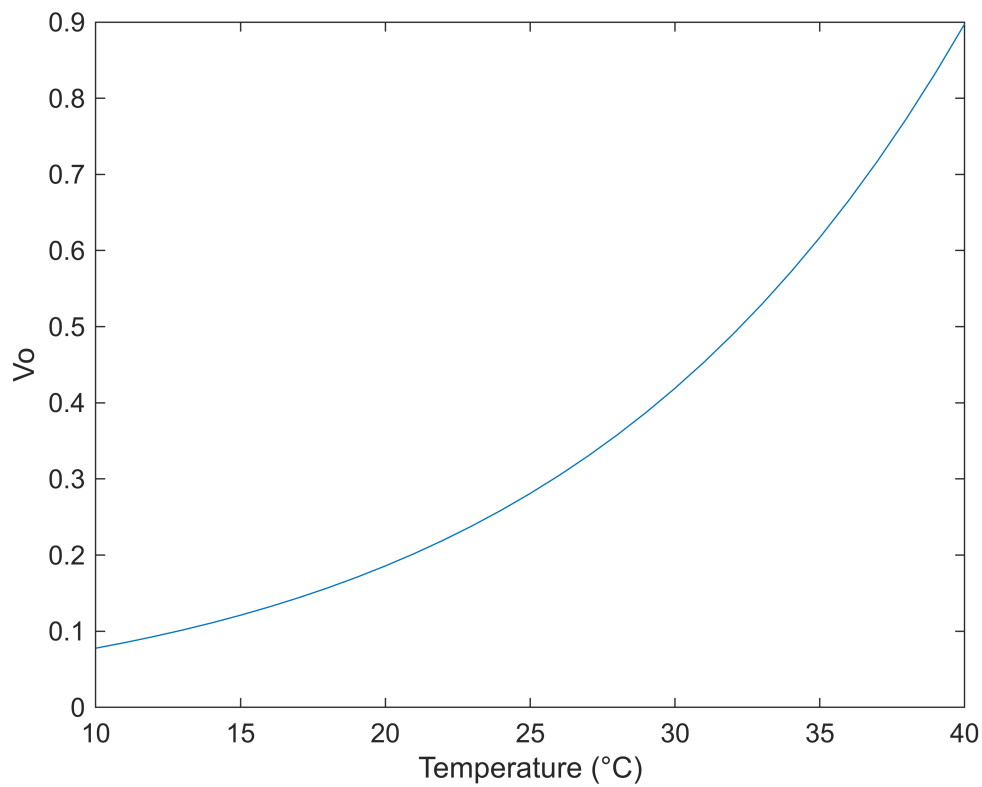
```
figure
Vo_plot=plot(Temp_Range(:,1),Vo_Vc_matrix(:,4))
```

Vo_plot =

Line with properties:

```

    Color: [0 0.4470 0.7410]
    LineStyle: '-'
    LineWidth: 0.5000
    Marker: 'none'
    MarkerSize: 6
    MarkerFaceColor: 'none'
    XData: [10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40]
    YData: [0.0778 0.0851 0.0930 0.1016 0.1110 0.1211 0.1321 0.1440 0.1568 0.1708 0.1858 0.2021 0.2196 0.2381 0.2575 0.2778 0.2990 0.3211 0.3440 0.3678 0.3925 0.4181 0.4446 0.4719 0.4999 0.5286 0.5579 0.5878 0.6183 0.6494 0.6811 0.7134 0.7463 0.7798 0.8138 0.8483 0.8833 0.9188 0.9548 0.9913 1.0283 1.0658 1.1037 1.1421 1.1809 1.2201 1.2597 1.2997 1.3400 1.3807 1.4218 1.4632 1.5050 1.5471 1.5896 1.6324 1.6756 1.7191 1.7629 1.8070 1.8514 1.8961 1.9411 1.9864 2.0320 2.0779 2.1241 2.1706 2.2174 2.2645 2.3119 2.3596 2.4076 2.4559 2.5044 2.5532 2.6023 2.6516 2.7012 2.7511 2.8012 2.8516 2.9022 2.9530 3.0041 3.0554 3.1069 3.1587 3.2107 3.2629 3.3154 3.3681 3.4211 3.4743 3.5277 3.5814 3.6353 3.6894 3.7437 3.7982 3.8529 3.9078 3.9629 4.0182 4.0737 4.1294 4.1853 4.2414 4.2977 4.3542 4.4109 4.4678 4.5249 4.5822 4.6397 4.6974 4.7553 4.8134 4.8717 4.9302 4.9889 5.0478 5.1069 5.1662 5.2257 5.2854 5.3453 5.4054 5.4657 5.5262 5.5869 5.6478 5.7089 5.7701 5.8315 5.8931 5.9549 6.0169 6.0791 6.1415 6.2041 6.2669 6.3299 6.3931 6.4565 6.5201 6.5839 6.6479 6.7121 6.7765 6.8411 6.9059 6.9709 7.0361 7.1015 7.1671 7.2329 7.2989 7.3651 7.4315 7.4981 7.5649 7.6319 7.6991 7.7665 7.8341 7.9019 7.9699 8.0381 8.1065 8.1751 8.2439 8.3129 8.3821 8.4515 8.5211 8.5909 8.6609 8.7311 8.8015 8.8721 8.9429 9.0139 9.0851 9.1565 9.2281 9.2999 9.3719 9.4441 9.5165 9.5891 9.6619 9.7349 9.8081 9.8815 9.9551 10.0289 10.1029 10.1771 10.2515 10.3261 10.4009 10.4759 10.5511 10.6265 10.7021 10.7779 10.8539 10.9301 11.0065 11.0831 11.1599 11.2369 11.3141 11.3915 11.4691 11.5469 11.6249 11.7031 11.7815 11.8601 11.9389 12.0179 12.0971 12.1765 12.2561 12.3359 12.4159 12.4961 12.5765 12.6571 12.7379 12.8189 12.8999 12.9811 13.0625 13.1441 13.2259 13.3079 13.3899 13.4721 13.5545 13.6371 13.7199 13.8029 13.8861 13.9695 14.0531 14.1369 14.2209 14.3051 14.3895 14.4741 14.5589 14.6439 14.7291 14.8145 14.8999 14.9855 15.0713 15.1573 15.2435 15.3299 15.4165 15.5033 15.5903 15.6775 15.7649 15.8525 15.9403 16.0283 16.1165 16.2049 16.2935 16.3823 16.4713 16.5605 16.6499 16.7395 16.8293 16.9193 17.0095 17.0999 17.1905 17.2813 17.3723 17.4635 17.5549 17.6465 17.7383 17.8303 17.9225 18.0149 18.1075 18.1999 18.2925 18.3853 18.4783 18.5715 18.6649 18.7585 18.8523 18.9463 19.0405 19.1349 19.2295 19.3243 19.4193 19.5145 19.6099 19.7055 19.8013 19.8973 19.9935 20.0899 20.1865 20.2833 20.3803 20.4775 20.5749 20.6725 20.7703 20.8683 20.9665 21.0649 21.1635 21.2623 21.3613 21.4605 21.5599 21.6595 21.7593 21.8593 21.9595 22.0599 22.1605 22.2613 22.3623 22.4635 22.5649 22.6665 22.7683 22.8703 22.9725 23.0749 23.1775 23.2803 23.3833 23.4865 23.5899 23.6935 23.7973 23.9013 24.0055 24.1099 24.2145 24.3193 24.4243 24.5295 24.6349 24.7405 24.8463 24.9523 25.0585 25.1649 25.2715 25.3783 25.4853 25.5925 25.6999 25.8075 25.9153 26.0233 26.1315 26.2399 26.3485 26.4573 26.5663 26.6755 26.7849 26.8945 27.0043 27.1143 27.2245 27.3349 27.4455 27.5563 27.6673 27.7785 27.8899 27.9915 28.1033 28.2153 28.3275 28.4399 28.5525 28.6653 28.7783 28.8915 29.0049 29.1185 29.2323 29.3463 29.4605 29.5749 29.6895 29.8043 29.9193 30.0345 30.1499 30.2655 30.3813 30.4973 30.6135 30.7299 30.8465 30.9633 31.0803 31.1975 31.3149 31.4325 31.5503 31.6683 31.7865 31.9049 32.0235 32.1423 32.2613 32.3805 32.4999 32.6195 32.7393 32.8593 32.9795 33.0999 33.2205 33.3413 33.4623 33.5835 33.7049 33.8265 33.9483 34.0699 34.1917 34.3137 34.4359 34.5583 34.6809 34.8037 34.9267 35.0499 35.1733 35.2969 35.4207 35.5447 35.6689 35.7933 35.9179 36.0427 36.1677 36.2929 36.4183 36.5439 36.6697 36.7957 36.9219 37.0483 37.1749 37.3017 37.4287 37.5559 37.6833 37.8109 37.9387 38.0667 38.1949 38.3233 38.4519 38.5807 38.7097 38.8389 38.9683 39.0979 39.2277 39.3577 39.4879 39.6183 39.7489 39.8797 40.0107 40.1419 40.2733 40.4049 40.5367 40.6687 40.8009 40.9333 41.0659 41.1987 41.3317 41.4649 41.5983 41.7319 41.8657 41.9997 42.1339 42.2683 42.4029 42.5377 42.6727 42.8079 42.9433 43.0789 43.2147 43.3507 43.4869 43.6233 43.7599 43.8967 44.0337 44.1709 44.3083 44.4459 44.5837 44.7217 44.8599 44.9983 45.1369 45.2757 45.4147 45.5539 45.6933 45.8329 45.9727 46.1127 46.2529 46.3933 46.5339 46.6747 46.8157 46.9569 47.0983 47.2399 47.3817 47.5237 47.6659 47.8083 47.9509 48.0937 48.2367 48.3799 48.5233 48.6669 48.8107 48.9547 49.0989 49.2433 49.3879 49.5327 49.6777 49.8229 49.9683 50.1139 50.2597 50.4057 50.5519 50.6983 50.8449 50.9917 51.1387 51.2859 51.4333 51.5809 51.7287 51.8767 52.0249 52.1733 52.3219 52.4707 52.6197 52.7689 52.9183 53.0679 53.2177 53.3677 53.5179 53.6683 53.8189 53.9697 54.1207 54.2719 54.4233 54.5749 54.7267 54.8787 55.0309 55.1833 55.3359 55.4887 55.6417 55.7949 55.9483 56.1019 56.2557 56.4097 56.5639 56.7183 56.8729 57.0277 57.1827 57.3379 57.4933 57.6489 57.8047 57.9607 58.1169 58.2733 58.4299 58.5867 58.7437 58.9009 59.0583 59.2159 59.3737 59.5317 59.6899 59.8483 60.0069 60.1657 60.3247 60.4839 60.6433 60.8029 60.9627 61.1227 61.2829 61.4433 61.6039 61.7647 61.9257 62.0869 62.2483 62.4099 62.5717 62.7337 62.8959 63.0583 63.2209 63.3837 63.5467 63.7099 63.8733 64.0369 64.1999 64.3633 64.5269 64.6907 64.8547 65.0189 65.1833 65.3479 65.5127 65.6777 65.8429 66.0083 66.1739 66.3397 66.5057 66.6719 66.8383 67.0049 67.1717 67.3387 67.5059 67.6733 67.8409 68.0087 68.1767 68.3449 68.5133 68.6817 68.8503 69.0191 69.1881 69.3573 69.5267 69.6963 69.8661 70.0361 70.2063 70.3767 70.5473 70.7181 70.8891 71.0603 71.2317 71.4033 71.5751 71.7471 71.9193 72.0917 72.2643 72.4371 72.6099 72.7829 72.9561 73.1295 73.3031 73.4769 73.6509 73.8251 73.9995 74.1741 74.3489 74.5239 74.6991 74.8745 75.0499 75.2255 75.4013 75.5773 75.7535 75.9299 76.1065 76.2833 76.4603 76.6375 76.8149 76.9925 77.1703 77.3483 77.5265 77.7049 77.8835 78.0623 78.2413 78.4205 78.5999 78.7795 78.9593 79.1393 79.3195 79.4999 79.6805 79.8613 80.0423 80.2235 80.4049 80.5865 80.7683 80.9503 81.1325 81.3149 81.4975 81.6803 81.8633 82.0465 82.2299 82.4135 82.5973 82.7813 82.9655 83.1499 83.3345 83.5193 83.7043 83.8895 84.0749 84.2605 84.4463 84.6323 84.8185 85.0049 85.1915 85.3783 85.5653 85.7525 85.9399 86.1275 86.3153 86.5033 86.6915 86.8799 87.0685 87.2573 87.4463 87.6355 87.8249 88.0145 88.2043 88.3943 88.5845 88.7749 88.9655 89.1563 89.3473 89.5385 89.7299 89.9215 90.1133 90.3053 90.4975 90.6899 90.8825 91.0753 91.2683 91.4615 91.6549 91.8485 92.0423 92.2363 92.4305 92.6249 92.8195 93.0143 93.2093 93.4045 93.5999 93.7955 93.9913 94.1873 94.3835 94.5799 94.7765 94.9733 95.1703 95.3675 95.5649 95.7625 95.9603 96.1583 96.3565 96.5549 96.7535 96.9523 97.1513 97.3505 97.5499 97.7495 97.9493 98.1493 98.3495 98.5499 98.7505 98.9513 99.1523 99.3535 99.5549 99.7565 99.9583 100.1603 100.3625 100.5649 100.7675 100.9703 101.1733 101.3765 101.5799 101.7835 101.9873 102.1913 102.3955 102.5999 102.8045 103.0093 103.2143 103.4195 103.6249 103.8305 104.0363 104.2423 104.4485 104.6549 104.8615 105.0683 105.2753 105.4825 105.6899 105.8975 106.1053 106.3133 106.5215 106.7299 106.9385 107.1473 107.3563 107.5655 107.7749 107.9845 108.1943 108.4043 108.6145 108.8249 109.0355 109.2463 109.4573 109.6685 109.8799 110.0915 110.3033 110.5153 110.7275 110.9399 111.1525 111.3653 111.5783 111.7915 112.0049 112.2185 112.4323 112.6463 112.8605 113.0749 113.2895 113.5043 113.7193 113.9345 114.1499 114.3655 114.5813 114.7973 115.0135 115.2299 115.4465 115.6633 115.8803 116.0975 116.3149 116.5325 116.7503 116.9683 117.1865 117.4049 117.6235 117.8423 118.0613 118.2805 118.4999 118.7195 118.9393 119.1593 119.3795 119.5999 119.8205 120.0413 120.2623 120.4835 120.7049 120.9265 121.1483 121.3703 121.5925 121.8149 122.0375 122.2603 122.4833 122.7065 122.9299 123.1535 123.3773 123.6013 123.8255 124.0499 124.2745 124.4993 124.7243 124.9495 125.1749 125.4005 125.6263 125.8523 126.0785 126.3049 126.5315 126.7583 126.9853 127.2125 127.4399 127.6675 127.8953 128.1233 128.3515 128.5799 128.8085 129.0373 129.2663 129.4955 129.7249 129.9545 130.1843 130.4143 130.6445 130.8749 131.1055 131.3363 131.5673 131.7985 132.0299 132.2615 132.4933 132.7253 132.9575 133.1899 133.4225 133.6553 133.8883 134.1215 134.3549 134.5885 134.8223 135.0563 135.2905 135.5249 135.7595 135.9943 136.2293 136.4645 136.6999 136.9355 137.1713 137.4073 137.6435 137.8799 138.1165 138.3533 138.5903 138.8275 139.0649 139.3025 139.5403 139.7783 140.0165 140.2549 140.4935 140.7323 140.9713 141.2105 141.4499 141.6895 141.9293 142.1693 142.4095 142.6499 142.8905 143.1313 143.3723 143.6135 143.8549 144.0965 144.3383 144.5803 144.8225 145.0649 145.3075 145.5503 145.7933 146.0365 146.2799 146.5235 146.7673 147.0113 147.2555 147.4999 147.7445 147.9893 148.2343 148.4795 148.7249 148.9705 149.2163 149.4623 149.7085 149.9549 150.2015 150.4483 150.6953 150.9425 151.1899 151.4375 151.6853 151.9333 152.1815 152.4299 152.6785 152.9273 153.1763 153.4255 153.6749 153.9245 154.1743 154.4243 154.6745 154.9249 155.1755 155.4263 155.6773 155.9285 156.1799 156.4315 156.6833 156.9353 157.1875 157.4399 157.6925 157.9453 158.1983 158.4515 158.7049 158.9585 159.2123 159.4663 159.7205 159.9749 160.2295 160.4843 160.7393 160.9945 161.2499 161.5055 161.7613 162.0173 162.2735 162.5299 162.7865 163.0433 163.3003 163.5575 163.8149 164.0725 164.3303 164.5883 164.8465 165.1049 165.3635 165.6223 165.8813 166.1405 166.3999 166.6595 166.9193 167.1793 167.4395 167.6999 167.9605 168.2213 168.4823 168.7435 169.0049 169.2665 169.5283 169.7903 170.0525 170.3149 170.5775 170.8403 171.1033 171.3665 171.6299 171.8935 172.1573 172.4213 172.6855 172.9499 173.2145 173.4793 173.7443 174.0095 174.2749 174.5405 174.8063 175.0723 175.3385 175.6049 175.8715 176.1383 176.4053 176.6725 176.9399 177.2075 177.4753 177.7433 178.0115 178.2799 178.5485 178.8173 179.0863 179.3555 179.6249 179.8945 180.1643 180.4343 180.7045 180.9749 181.2455 181.5163 181.7873 182.0585 182.3299 182.6015 182.8733 183.1453 183.4175 183.6899 183.9625 184.2353 184.5083 184.7815 185.0549 185.3285 185.6023 185.8763 186.1505 186.4249 186.6995 186.9743 187.2493 187.5245 187.7999 188.0755 188.3513 188.6273 188.9035 189.1799 189.4565 189.7333 190.0103 190.2875 190.5649 190.8425 191.1203 191.3983 191.6765 191.9549 192.2335 192.5123 192.7913 193.0705 193.3499 193.6295 193.9093 194.1893 194.4695 194.7499 195.0305 195.3113 195.5923 195.8735 196.1549 196.4365 196.7183 196.9999 197.2817 197.5637 197.8459 198.1283 198.4109 198.6937 198.9767 199.2599 199.5433 199.8269 200.1107 200.3947 200.6789 200.9633 201.2479 201.5327 201.8177 202.1029 202.3883 202.6739 202.9597 203.2457 203.5319 203.8183 204.1049 204.3917 204.6787 204.9659 205.2533 205.5409 205.8287 206.1167 206.4049 206.6933 206.9819 207.2707 207.5597 207.8489 208.1383 208.4279 208.7177 209.0077 209.2979 209.5883 209.8789 210.1697 210.4607 210.7519 211.0433 211.3349 211.6267 211.9187 212.2109 212.5033 212.7959 213.0887 213.3817 213.6749 213.9683 214.2619 214.5557 214.8497 215.1439 215.4383 215.7329 216.0277 216.3227 216.6179 216.9133 217.2089 217.5047 217.8007 218.0969 218.3933 218.6899 218.9867 219.2837 219.5809 219.8783 220.1759 220.4737 220.7717 221.0699 221.3683 221.6669 221.9657 222.2647 222.5639 222.8633 223.1629 223.4627 223.7627 224.0629 224.3633 224.6639 224.9647 225.2657 225.5669 225.8683 226.1699 226.4717 226.7737 227.0759 227.3783 227.6809 227.9837 228.2867 228.5899 228.8933 229.1969 229.5007 229.8047 230.1089 230.4133 230.7179 231.0227 231.3277 231.6329 231.9383 232.2439 232.5497 232.8557 233.1619 233.4683 233.7749 234.0817 234.3887 234.6959 235.0033 235.3109 235.6187 235.9267 236.2349 236.5433 236.8519 237.1607 237.4697 237.7789 238.0883 238.3979 238.7077 239.0177 239.3279 239.6383 239.9489 240.2597 240.5707 240.8819 241.1933 241.5049 241.8167 242.1287 242.4409 242.7533 243.0659 243.3787 243.6917 244.0049 244.3183 244.6319 244.9457 245.259
```



```
figure
Vc_plot=plot(Temp_Range(:,1),Vo_Vc_matrix(:,3))
```

```
Vc_plot =
```

```
Line with properties:
```

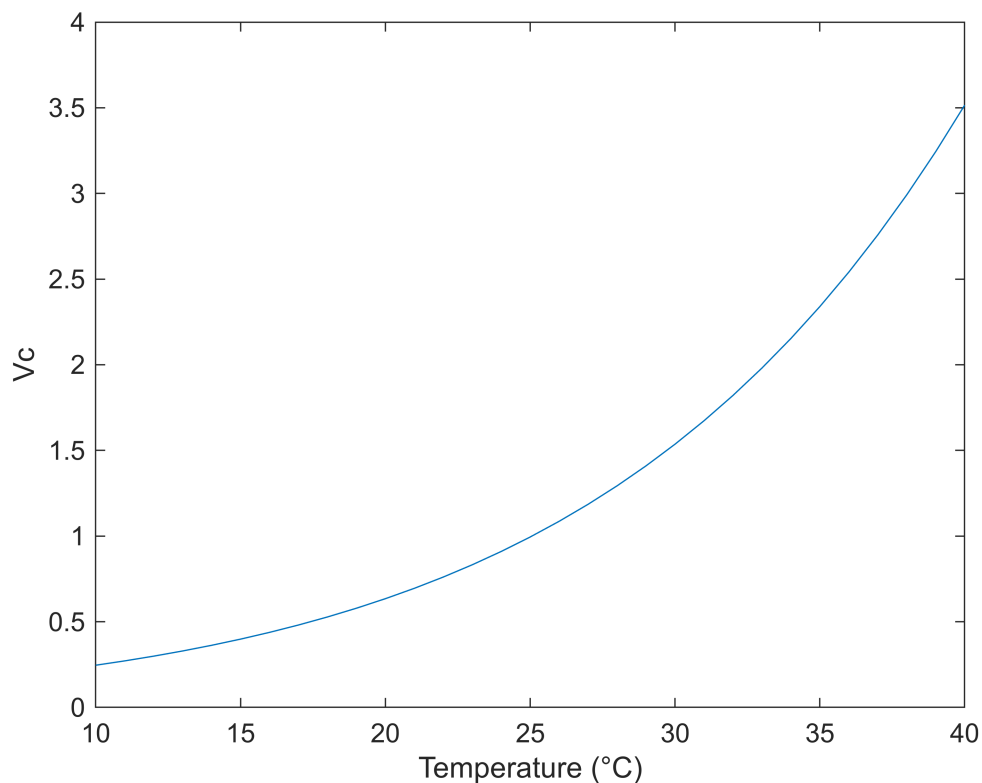
```

    Color: [0 0.4470 0.7410]
    LineStyle: '-'
    LineWidth: 0.5000
    Marker: 'none'
    MarkerSize: 6
    MarkerFaceColor: 'none'
    XData: [10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40]
    YData: [0.2462 0.2715 0.2991 0.3294 0.3624 0.3985 0.4380 0.4810 0.5279 0.5789 0.6346 0.6951 0.7610 0.8320 0.9000]
```

```
Show all properties
```

```

xlabel('Temperature (°C)')
ylabel('Vc')
set(gcf, 'PaperOrientation', 'landscape');
print(gcf,fullfile('Outputs/rice_params/graphs',"Temp_vs_Vc"),'-djpeg');
```



Calculate V_c and $\frac{V_o}{V_c}$ ratio for Bernacchi at measurement temperature T_p :

```
Bernacchi_Vc_Tp = exp(c_Vc - dHa_Vc / (R * Tp));
Bernacchi_Vo_Tp = exp(c_Vo - dHa_Vo / (R * Tp));
Bernacchi_PrPs_ratio_Tp = Bernacchi_Vo_Tp/Bernacchi_Vc_Tp;
```

Work out the $\frac{V_o}{V_c}$ ratio for rice at 25°C using data in Makino et al (1988):

```
Makino_Vc_25 = 1.77; % RuBP carboxylase activity at pH 8.15 and 25°C from rice
leaves (umol(mg enzyme)-1 min-1 (Vmax)
Makino_Vo_25 = 0.58; % RuBP oxygenase activity at pH 8.15 and 25°C from rice leaves
(umol(mg enzyme)-1 min-1 (Vmax)
Makino_PrPs_ratio_25 = Makino_Vo_25/Makino_Vc_25; % Check this matches the value in
von Caemmerer (2000)
```

Adjusting the value V_{cmax} at 25°C to measurement temperature T_p

We can use one of two approaches to calculate the Makino value of $\frac{V_o}{V_c}$ at T_p :

1. Use the ratios of Makino and Bernacchi at 25°C and multiply by Bernacchi parameter estimate at a given temperature
2. Fit an Arrhenius equation to a non-linear model to find the Bernacchi temperature response of $\frac{V_o}{V_c}$ (plotted above)

Use linear scaling of Makino to Bernacchi ratios:

```
Makino_PrPs_ratio_Tp = Makino_PrPs_ratio_25*Bernacchi_PrPs_ratio_Tp/
Bernacchi_PrPs_ratio_25
```

```
Makino_PrPs_ratio_Tp = 0.3188
```

```
%OR
Makino_PrPs_ratio_Tp2 = Makino_PrPs_ratio_25/
Bernacchi_PrPs_ratio_25*Bernacchi_PrPs_ratio_Tp
```

```
Makino_PrPs_ratio_Tp2 = 0.3188
```

Non-linear fits for $\frac{V_o}{V_c}$

Define predictor (T) and response (y) variables:

```
T = Vo_Vc_matrix(:,2); % Temp in K
y = Vo_Vc_matrix(:,5); % Vo/Vc ratio
```

So far, we have defined the Arrhenius equation as:

$$\text{Parameter} = \exp \left(c - \frac{\Delta H_a}{RT_K} \right)$$

c is meant to scale the amplitude of the function whereas dHa represents the temperature dependence relationship.

Using term c-dHa combined these two effects into one, making them harder to interpret individually.

Without c, the model assumes that dHa depends solely on temperature and ignores other potential system-specific variations.

This means the equation can also be rewritten as:

$$y = c \cdot \exp \left(- \frac{\Delta H_a}{RT_K} \right)$$

to ensure that the exponent has a negative sign - previously $\exp(c-dHa/R*T)$ was causing errors since the exponent was positive.

Define an exponential model with non-linear parameters using this form of the Arrhenius equation:

```

arrhenius_model = fittype('c * exp(-dHa ./ (R * T))', ...
    'independent', 'T', ...
    'coefficients', {'c', 'dHa'}, ...
    'problem', 'R');

```

Fit the model using the fit function and passing in the model structure with initial guesses for c and dHa:

```

fittedModel = fit(T(:), y(:), arrhenius_model, 'StartPoint', [0, 2], 'problem', R);

```

Display a summary of the fit:

```

disp(fittedModel)

```

```

General model:
fittedModel(T) = c * exp(-dHa ./ (R * T))
Coefficients (with 95% confidence bounds):
    c =      0.03439   (0.03439, 0.03439)
    dHa =      -5.22   (-5.22, -5.22)
Problem parameters:
    R =      0.008314

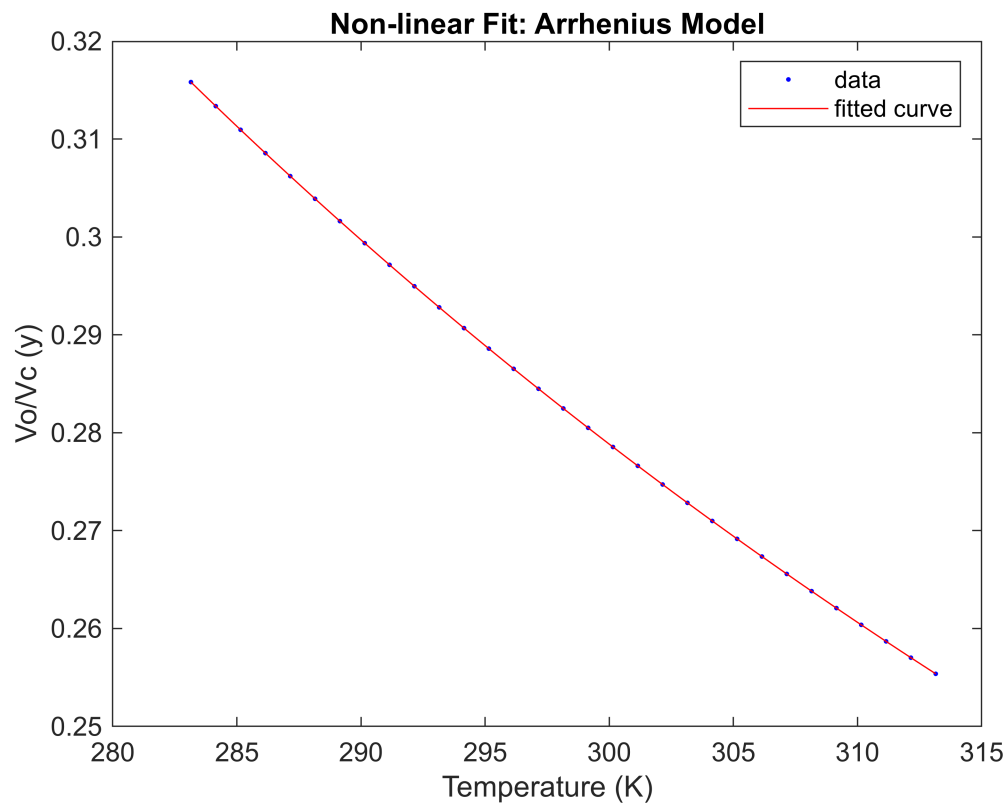
```

Plot the non-linear fit of the model to the data:

```

figure;
plot(fittedModel, T, y);
xlabel('Temperature (K)');
ylabel('Vo/Vc (y)');
title('Non-linear Fit: Arrhenius Model');

```



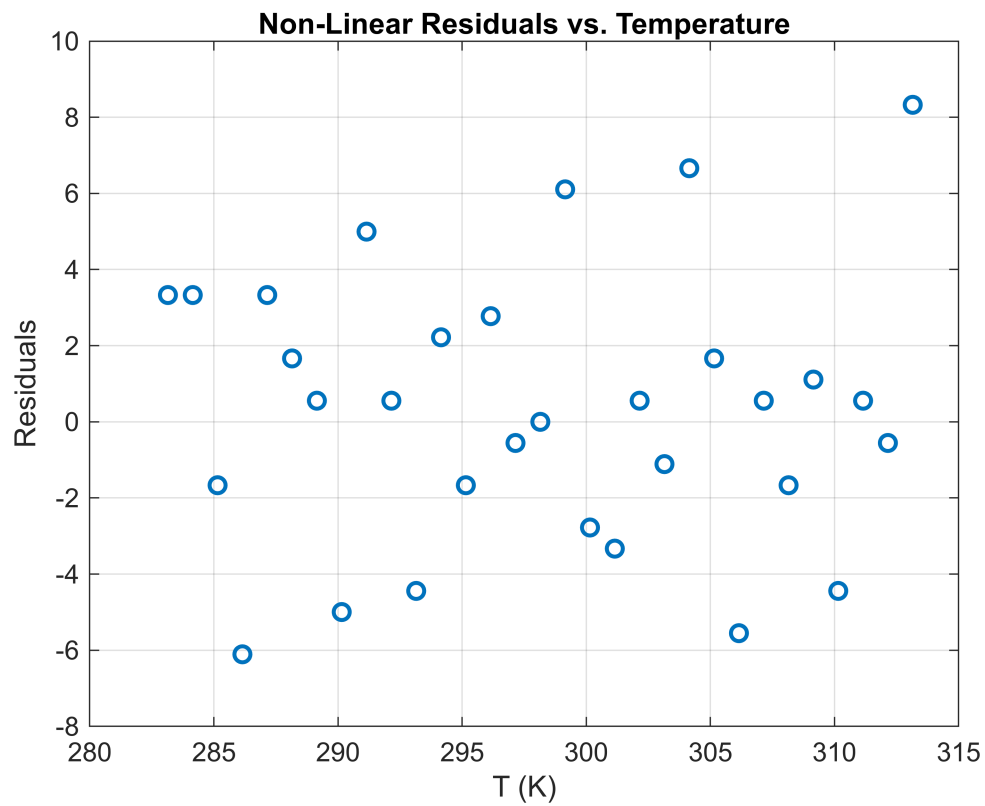
Calculate the residuals:

```
% Specify predicted values from the model
y_fit = fittedModel(T);

% Calculate residuals by subtracting from the observed y values
residuals = y - y_fit; % Observed - Predicted
```

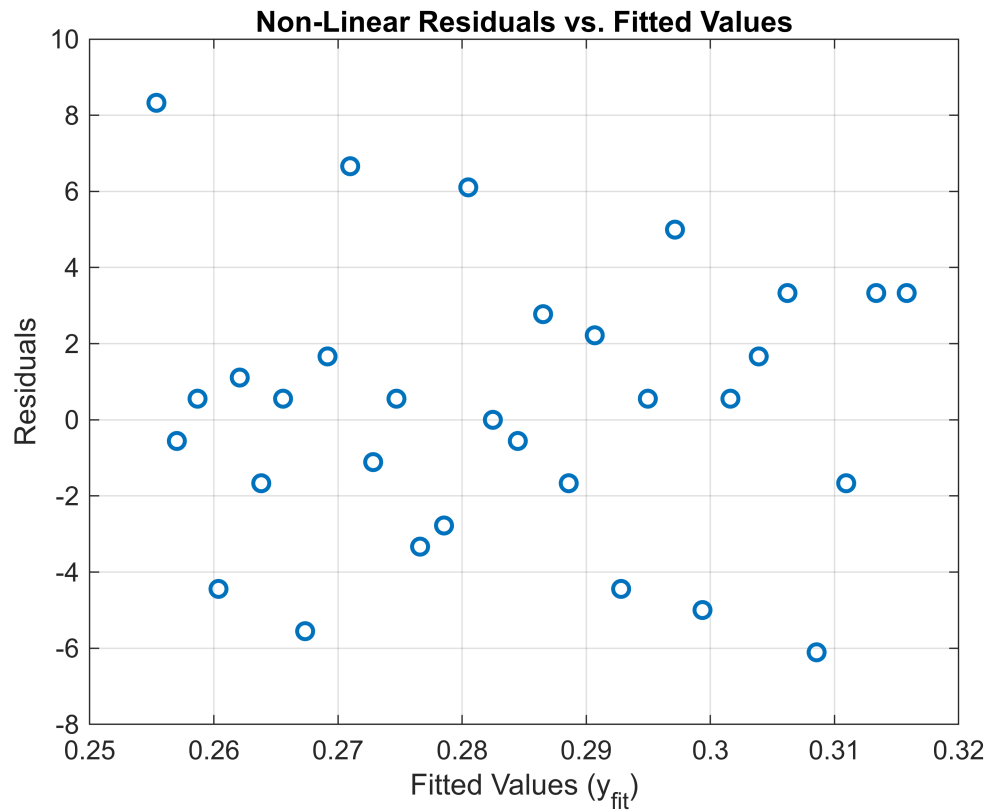
Plot residuals against T:

```
figure;
plot(T, residuals, 'o', 'LineWidth', 1.5);
xlabel('T (K)');
ylabel('Residuals');
title('Non-Linear Residuals vs. Temperature');
grid on;
```

Plot residuals against fitted values:

```
figure;  
plot(y_fit, residuals, 'o', 'LineWidth', 1.5);  
xlabel('Fitted Values (y_{fit})');  
ylabel('Residuals');  
title('Non-Linear Residuals vs. Fitted Values');  
grid on;
```



Random scatter around zero indicates we have a good fit of model to data.

The values of the residuals are also very small ($< 10 \times 10^{-16}$), indicating the model accounts for all variations in the data and there are no unexplained residuals left.

Calculate R-squared:

```
SST = sum((y - mean(y)).^2)    % Total sum of squares
```

```
SST = 0.0100
```

```
SSR = sum(residuals.^2)        % Residual sum of squares
```

```
SSR = 3.9166e-30
```

```
R_squared = 1 - (SSR / SST)
```

```
R_squared = 1
```

Check confidence intervals:

```
conf_int = confint(fittedModel)
```

```
conf_int = 2x2
    0.0344    -5.2200
    0.0344    -5.2200
```

The R-squared is exactly 1, which indicates all the data points lie on the fitted curve perfectly - this should be treated with caution as it could indicate overfitting along with the fact that the confidence bounds for c and dHa were exactly the same as their estimated values.

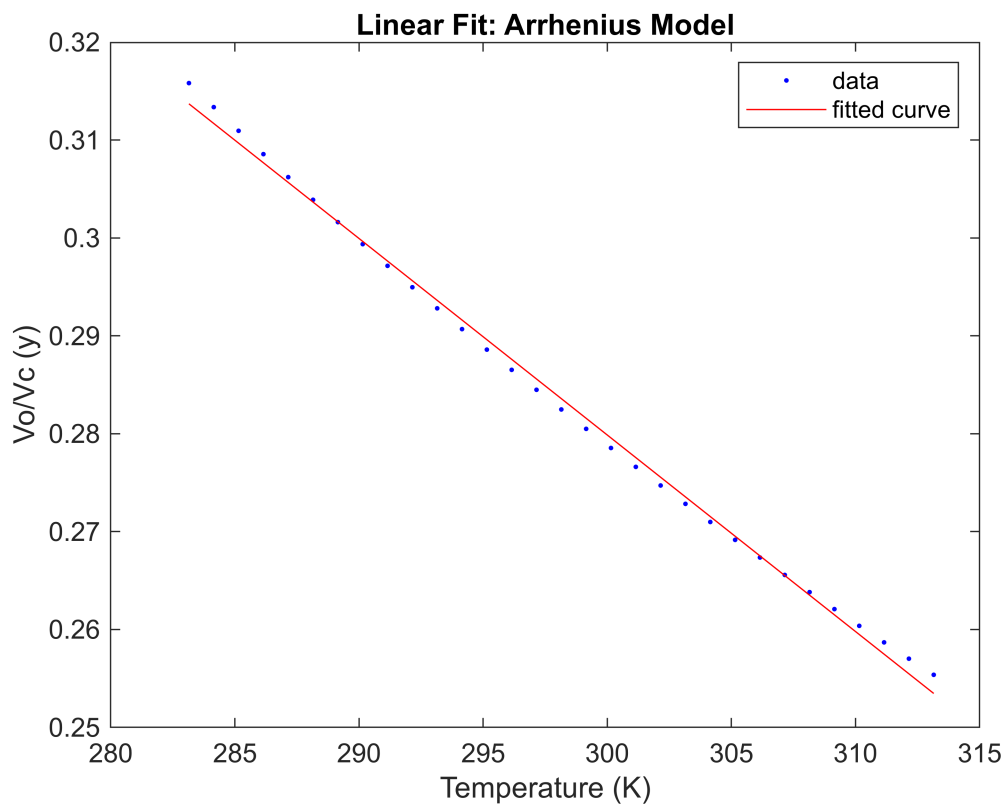
Fit a linear model for comparison to this:

```
linear_model = fit(T(:), y(:), 'poly1');  
y_linear_fit = linear_model(T);  
disp(linear_model)
```

```
Linear model Poly1:  
linear_model(x) = p1*x + p2  
Coefficients (with 95% confidence bounds):  
p1 = -0.002008 (-0.00205, -0.001966)  
p2 = 0.8822 (0.8696, 0.8947)
```

Plot the linear model:

```
figure;  
plot(linear_model, T, y);  
xlabel('Temperature (K)');  
ylabel('Vo/Vc (y)');  
title('Linear Fit: Arrhenius Model');
```



Compare R-squared values between linear and non linear:

```
SSR_linear = sum((y - y_linear_fit).^2)
```

```
SSR_linear = 3.0392e-05
```

```
R2_linear = 1 - (SSR_linear / SST)
```

```
R2_linear = 0.9970
```

```
R2_complex = 1 - (SSR / SST)
```

```
R2_complex = 1
```

The simpler linear model performs almost as well as the more complex non-linear model, so the non-linear may have been overfitting.

However, this might not matter if we only need the Vo/Vc for values between 10 and 40° C and not generalising to values outside this dataset.