```
Sub DTAsim()
'DTA-Simulation

Dim z0 As Integer      'Zählindex für Startzeile
Dim z As Long          'Zählindex für Zeile
Dim ze As Long         'Anzahl der Sützstellen (Zeitzeilen)
Dim c As Integer       'Zählindex für Spalte
Dim i As Integer       'Zählindex für Temperaturen
Dim n As Integer       'Anzahl der Phasen
Dim u(3) As Double   'deltaT
Dim ti As Long         'time, s
'Dim ti0 As Double     'startzeit
'Dim tie As Double     'endzeit
Dim T As Double        'Temperatur,°C
Dim T0 As Double       'Starttemperatur,°C
Dim Te As Double       'Endtemperatur, °C
Dim s As Single        'Schrittweite Zeit, s
Dim c0 As Double       'Konstanten für Thermoelement
Dim c1 As Double       '
Dim c2 As Double       '
Dim c3 As Double       '
Dim c4 As Double       '
Dim c5 As Double       '
Dim c6 As Double       '
Dim c7 As Double       '
Dim c8 As Double       '
Dim EMK As Double                'EMK Thermoelement
Dim MS As Double                 'Sample mass
Dim alphaminutes As Single  'Heating Rate alpha for min
Dim alpha As Double              'Heating rate/second
Dim mcrucible As Double
Dim msample As Double
Dim MC As Double
Dim a As Double                  'Zeitkonszanten
Dim b As Double
Dim d As Double
Dim R As Double
Dim dH As Double                 'Ableitung Enthalpie
Dim lngLR As Long
Dim K As Double                  'T in K
Dim MC1 As Double                'Wärmekapazität Tiegel
Dim MC2 As Double
Dim MA As Single                 'mittlere Wärmekapazität Sample
```

```
'Deklaration der Vektoren, Matrizen und Indizes
Dim y(3) As Double              ' Vektor der Temperaturen
Dim abltg(4, 3) As Double   ' Matrix der Ableitungen
Dim du(4, 3) As Double         ' Matrix der differenziellen Konzentrationsänderungen
Dim rz As Integer                 ' Zählindex für Runge-Kutta-Iteration (rz = 1..4)
Dim p As Double
Dim q As Double



'Löschen des Arbeitsblattes "Diagram" (falls vorhanden)

Application.DisplayAlerts = False   ' Unterdrückt die Fehleranzeige,
                              ' falls Arbeitsblatt "Diagram" nicht vorhanden
On Error Resume Next                  ' und lässt das Programm weiterlaufen
Sheets("Diagram").Delete              ' Falls vorhanden: wird gelöscht
Application.DisplayAlerts = True       ' Fehleranzeige wird wieder aktiviert

'Löschen überflüssiger Arbeitsblätter "Calculation" (falls vorhanden)

Application.DisplayAlerts = False
While Worksheets.Count > 1
Worksheets(2).Delete
Wend
Application.DisplayAlerts = True

'Einlesen der Werte
ThisWorkbook.Sheets(1).Activate

'Rows("2:65536").Select
'Selection.NumberFormat = "0.000000"

T0 = InputBox("Temperature Start [°C]")
'Range("g3", "g4").Value = T0

Te = InputBox("Temperature End [°C]")
'Range("g4").Value = Te

MA = InputBox("Average molar mass sample [g/mol]")

'Umsortierung der Temperaturewerte aufsteigend oder absteigend
```

```
If T0 > Te Then
Worksheets(1).Select
lngLR = IIf(IsEmpty(Cells(Rows.Count, 6)), Cells(Rows.Count, 6).End(xlUp).Row, Rows.Count)
Range("E3", Cells(lngLR, 6)).Sort Key1:=Range("E3"), Order1:=xlDescending
Worksheets("Calculation").Select

ElseIf T0 < Te Then
Worksheets(1).Select
lngLR = IIf(IsEmpty(Cells(Rows.Count, 6)), Cells(Rows.Count, 6).End(xlUp).Row, Rows.Count)
Range("E3", Cells(lngLR, 6)).Sort Key1:=Range("E3"), Order1:=xlAscending
Worksheets("Calculation").Select
End If

mcrucible = 0.539 'Mass of crucible in g
msample = 0.109 'Mass of sample in g


'alphaminutes = Range("c4").Value
'alphaminutes = 6       'die Anzahl der Enthalpiewerte pro minute
alphaminutes = InputBox("Heating Rate [K/min]")
alpha = alphaminutes / 60 'Schrittweite Temperatur
s = 1 '/ alpha                          'Schrittweite Zeit , s

ze = Abs(Te - T0) / alpha
'ze = Range("f6").Value
z0 = 2
n = 3

'EMK Thermoelement

c0 = 0
c1 = 5.40313308631 * 10 ^ -3
c2 = 1.2593428974 * 10 ^ -5
c3 = -2.32477968689 * 10 ^ -8
c4 = 3.22028823036 * 10 ^ -11
c5 = -3.31465196389 * 10 ^ -14
c6 = 2.55744251786 * 10 ^ -17
c7 = -1.25068871393 * 10 ^ -20
c8 = 2.71443176145 * 10 ^ -24

'Time constants of DSC apparatus

a = 12.5  ' (* a: sample to cup, Subscript[t, S,C](s) *)
```

```
b = 18.2  ' (* b: wall to cup, Subscript[t, W,C](s) *)
d = 0.82  '(* d: tc to cup, Subscript[t, T,C](s) *)
'R = 0.51 '(* R: average sample mass and solid/liquid heat capacity *)


Worksheets.Add(After:=Worksheets(Worksheets.Count)).Name = "Calculation"
ThisWorkbook.Sheets("Calculation").Activate

'Zellformatierung

Rows("1:65536").ClearContents
Rows("2:65536").Select

Selection.NumberFormat = "0.000000"

Range("A2:A65536").Select

Selection.NumberFormat = "0.000000"

'Zuweisung der Anfangstemperaturen

For c = n - 1 To n + 1
Worksheets("Calculation").Cells(1, 2).Value = "Tsam"
Worksheets("Calculation").Cells(1, 3).Value = "Tcup"
Worksheets("Calculation").Cells(1, 4).Value = "Ttc"
Worksheets("Calculation").Cells(2, c).Value = T0
Next c

'Generierung der Temperaturwerte

Worksheets("Calculation").Cells(1, 1).Value = "Temperatur, °C"

For z = 0 To ze  'Durchführung der Berechnungen

If T0 > Te Then
T = T0 - alpha * z

ElseIf T0 < Te Then
T = T0 + alpha * z

End If

K = T + 273.15
```

```
'MA = ((0.153 * 26.982) + (0.67 * 24.305) + (0.177 * 65.409))
'MC1 = (70.12 + 7.02 * 10 ^ -3 * K - 1.42 * 10 ^ 6 * K ^ -2) / 123.223 'ZrO2 (M. Binnewies, E.
Milke Thermochemical Data of Elements and Compounds)
'MC1 = 69 / 123.223  'ZrO2
'MC2 = 79.04 / 101.96 ' Al2O3
MC1 = (27.78 - 3.84 * 10 ^ -3 * K - 0.26 * 10 ^ 6 * K ^ -2 + 3.6 * 10 ^ -6 * K ^ 2) / 92.9 ' Niob
MC = (mcrucible / msample) * MC1

'Ableitung der Enthalpie Htot

Worksheets("Calculation").Cells(z + z0, 1).Value = Round(T, 5)
dH = Abs(((Worksheets(1).Cells(z + 4, 5).Value) / (MA)) - ((Worksheets(1).Cells(z + 3, 5).Value)
/ (MA))) / (alpha)
Worksheets(1).Cells(z + 3, 24).Value = Round(dH, 4)

ti = z + alpha '(Worksheets("Calculation").Cells(z + z0, 1).Value - T0) / alpha
For i = 1 To n
u(i) = Worksheets("Calculation").Cells(z + z0, i + 1).Value
Next i

Call ruku(u, i, n, y, rz, MC, dH, abltg, du, a, b, d, T, s, z, z0)

Next z

'Durchführung der Berechnungen

c = n + 3
Worksheets("Calculation").Cells(1, c - 1).Value = "T" 'Name
'Worksheets("Calculation").Cells(1, c).Value = "deltaT-dTmax" 'Name
Worksheets("Calculation").Cells(1, c).Value = "deltaT"  'Name
Worksheets("Calculation").Cells(1, c + 1).Value = "muV" 'Name

For z = 2 To ze

Worksheets("Calculation").Cells(z, c - 1).Value = Worksheets("Calculation").Cells(z, c - 5).Value
l = Cells(z + 2, c - 5)
o = Cells(z + 2, c - 2)
p = l - o

Worksheets("Calculation").Cells(z, c).Value = Round(p, 5)
Next z

'Umrechnung in muV
```

```
For z = 2 To ze
p = Worksheets("Calculation").Cells(z, c).Value
EMK = 1000 * (c0 + (c1 * p) + (c2 * p ^ 2) + (c3 * p ^ 3) + (c4 * p ^ 4) + (c5 * p ^ 5) + (c6 * p ^ 6)
+ (c7 * p ^ 7) + (c8 * p ^ 8))
Worksheets("Calculation").Cells(z, c + 1).Value = Round(EMK, 5)
Next z

'Erstellung des Diagrams

Call Diagram_erstellen

'Ende des Makros Calculation

MsgBox ("End of calculation")

End Sub

Sub ruku(u, i, n, y, rz, MC, dH, abltg, du, a, b, d, T, s, z, z0)

'-------------------------------------------------------------------------------------------
' Unterprogramm Ruku
' Führt die Berechnungen nach dem Runge-Kutta-Verfahren 4. Ordung mit fester
' Schrittweite durch
' Zuweisung der Temperaturen der Funktion y im RuKu-"Rezept"
' (siehe Blatt 1)
For i = 1 To n
y(i) = u(i)
Next i

' Abarbeiten der Runge-Kutta-Schleifen
For rz = 1 To 4
'y(1) = Worksheets("Calculation").Cells(z + z0, 2).Value 'Tsam =
Worksheets("Calculation").Cells(z + 2, 2).Value
'y(2) = Worksheets("Calculation").Cells(z + z0, 3).Value 'Tcup =
Worksheets("Calculation").Cells(z + 2, 3).Value
'y(3) = Worksheets("Calculation").Cells(z + z0, 4).Value 'Ttc = Worksheets("Calculation").Cells(z
+ 2, 4).Value

' Definition des Differenzialgleichungssystems (siehe Blatt 1)
abltg(rz, 1) = (MC) * (1 / dH) * (1 / a) * (y(2) - y(1)) 'dTsam
abltg(rz, 2) = ((1 / a) * (y(1) - y(2))) + ((1 / b) * (T - y(2))) 'dTcup
abltg(rz, 3) = (1 / d) * (y(2) - y(3)) 'dTtc
```

```
' Definition der differenziellen Temperaturänderungen
For i = 1 To n
du(rz, i) = abltg(rz, i) * s ' Entspricht h*p_i (siehe Blatt 1)
Next i
' Berechung von p_2 bis p_4
' (p_1 muss nicht berechnet werden, da es c_i entspricht!)
For i = 1 To n
Select Case rz ' Verzweigungsanweisung (rz = Prüfausdruck!),
' sprich: wenn rz=1, dann wird p_2 berechnet usw.
Case 1
y(i) = u(i) + du(rz, i) / 2 ' Entspricht p_2
Case 2
y(i) = u(i) + du(rz, i) / 2 ' Entspricht p_3
Case 3
y(i) = u(i) + du(rz, i) ' Entspricht p_4
' Berechnung der Konzentration
Case 4
u(i) = u(i) + (1 / 6) * (du(1, i) + 2 * du(2, i) + 2 * du(3, i) + du(4, i))
' Schreiben der Ergebnisse in entsprechende Zellen
' (z+z0 zusätzlich +1, da in der 2. Zeile bereits die Anfangsbedingungen stehen!)
Worksheets("Calculation").Cells(z + z0 + 1, i + 1).Value = Round(u(i), 5)
End Select ' Ende der Verzweigungsanweisung
Next i ' Berechnung für die nächsten Temperatur (nächste Spalte)
Next rz ' Berechnung für den nächsten Zählindex
' für Runge-Kutta-Iteration
'--------------------------------------------------------------------------------

End Sub

Public Sub Diagram_erstellen()
'
' Diagram_erstellen Makro
'
' Abkürzung: Ctrl+Shift+D
'
    Charts.Add
    ActiveChart.ChartType = xlXYScatterSmoothNoMarkers
    ActiveChart.SetSourceData Source:=Sheets("Calculation").Range("e1:g30002"), _
    PlotBy:=xlColumns
    ActiveChart.Location Where:=xlLocationAsNewSheet, Name:="Diagram"
    With ActiveChart
    .HasTitle = True
```

```
        .ChartTitle.Characters.Text = "DSC Simulation"
        .Axes(xlCategory, xlPrimary).HasTitle = True
        .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "Temperature [°C]"
        .Axes(xlValue, xlPrimary).HasTitle = True
        .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = _
        "delta T [°C] and DSC [muV]"
        End With
        With ActiveChart.Axes(xlCategory)
        .HasMajorGridlines = True
        .HasMinorGridlines = True
        End With
        With ActiveChart.Axes(xlValue)
        .HasMajorGridlines = True
        .HasMinorGridlines = True
        End With

End Sub
```