# Scilab-Programme zur Planck-Temperatur

Funktionsüberblick

Programme

- tempfunktion_entwickeln.sce
- tempfunktion_plotten.sce
- wav2gif_v5.sce

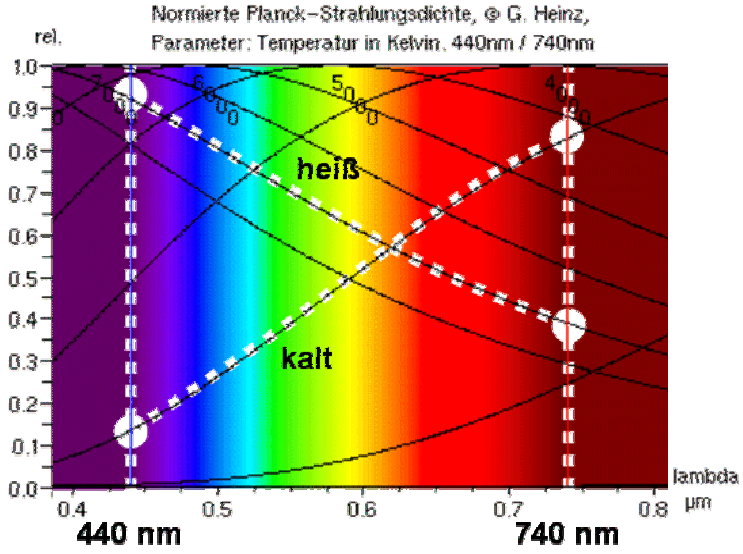
Version Scilab 3.1.1

- Scilab http://www.scilab.org/
- Download http://www.scilab.org/communities/developer_zone/scilab_versions/oldreleases/scilab_3.1.1

Impressum
Dr. G. Heinz, GFaI
Vollmerstr.3
12489 Berlin
www.gfai.de/~heinz
heinz@gfai.de
Mob. 0172 545 6036

# Funktionsüberblick

- Plancksche Schwarzkörperstrahlung wird vorausgesetzt
- Zwei gemessene Emissionswerte e1(λ1), e2(λ2) definieren eine Temperatur T über die Plankschen Strahlungsgleichung, z.B. in der Form siehe Bild
- Dann kann man schreiben:
  e1 = e1(T) und e2 = e2(T)
- Umgekehrt wird jede Temperaturkurve mit diesen zwei Punkten *e1, e2* definiert:
  T = T(e1,e2)
- Das Programm
  *tempfunktion_entwickeln.sce*
  generiert die Tempfkt. als Tabelle in einem *.bin-File (einmalig für eine Diodenpaarung)




- Mit *wav2gif_v5.sce* läßt sich aus je zwei Zeitfunktionen von Photodioden die Plancktemperatur als Zeitfunktion bestimmen
- Mit *tempfunktion_plotten.sce*
  kann eine vorhandene Temperaturfunktion eingelesen und geplottet werden

# Funktionsüberblick

- Die Plancksche Temperaturfunktion wird numerisch erzeugt und gespeichert als *.bin in einer Matrix vierer aus 4 Spalten und n Zeilen
- Vierer-Matrix:
  - 1.Spalte: Temperatur in Kelvin
  - 2.Spalte: Emissionswerte e1
  - 3.Spalte: Emissionswerte e2
  - 4.Spalte: Quotient e1/e2
- Beispiel: Test auf Scilab-Konsole --> vierer
- Werte aus Zeitfunktionen (*.wav) greifen über lineare Interpolation in diesen File, um die zugehörige Temperatur zu erhalten
- Zur Berechnung der Zeitfunktionen wird eine Quotientenform genutzt
- Der Quotient der Emissionswerte (e1/e2) entspricht einer Temperatur T
  T = T(e1/e2)

# tempfunktion_entwickeln.sce

Input:

- *.pin Textfile zur Parametereinstellung –
  wird bei Start im Dialog abgefragt

Output:

- *.bin Binärfile mit der Temperaturfunktion als
  Vierermatrix vierer=[grad,e1,e2,e1zue2]
  - » grad: Temperatur in Kelvin (Output)
  - » e1, e2: Emissionswerte (Input) (unbenutzt)
  - » e1zue2: Ratio e1 zu e2 als Input

- Scilab-Graphic(0) bis (3): Sichten auf die Temp.-Funktion, b.w.

- *.gif Bild der entwickelten Temperaturfunktion (3)

# tempfunktion_entwickeln.sce – Parameterinit *.pin

## Beispiel tempfkt_1950nm_2300nm.pin

```
//  Parameter-Init für Planck-Temperaturkurve entwickeln

// Bestimmungs-Wellenlängen:
      L1=1.95;     // Empfindlichkeitsmaximum erster Photodiode in µm
      L2=2.3;      // Empfindlichkeitsmaximum zweiter Photodiode in µm
      Ln=300;      // Anzahl der lambda-Stützpunkte pro Kurve
      Lstart=1.0;  // Von Wellenlänge [in µm]
      Lstop=3.0;   // Bis Wellenlänge [in µm]
      m=0.000001;  // Korrekturfaktor, wenn lambda in Mikrometer eingesetzt wird

// Temperaturen:
      Tanf=300;    // Beginnen bei Temperatur
      Tend=3000;   // Endtemperatur
      T=Tanf;      // Temperatur in Kelvin
      dT = 10     // Temperatur-Schrittweite für hübsches Bild
      //dT= (Tend-Tanf)/Ln ;     // Temperatur-Schrittweite für Ablage in *.bin-File
      tempk = 1;   // Absolut-Kalibrierung der Temperatur

// Strings:
      filename='tempfkt_'+string(L1*1e3)+'nm'+'_'+string(L2*1e3)+'nm'; // für Ausgabefiles
      col='black'; // Farbe des Ergebnisplots #3
```

# tempfunktion_entwickeln.sce - Scilab-Dialog

```
Approximation der Planck-Temperatur
aus dem Pegel zweier Spektrallinien

Autor: G. Heinz, GFaI-Berlin, 12/2007 - 1/2008
heinz@gfai.de, www.gfai.de/~heinz
Copyrights: Uneingeschränkte Nutzung erlaubt bei
        Quellenverweis.

Input:  Emissionspegel e1 and e2 bei Wellenlänge L1 and
        L2 in µm
Output: Temperaturfunktion T = f(e1/e2)

Absolutpegel von e1 und e2 sind unbekannt ->
        Kalibrierung mit tempk
 dT  =    100.

From parameter file:
C:\Daten\_Bibliothek\Bits_and_Bytes\Scilab\wav2gif_TU\
        tempfkt_1950nm_2300nm.pin

L1 = 1.950000 µm
L2 = 2.300000 µm
dT = 100.000000 K

... GIF-Plot of Window 0 saved as
        tempfkt_1950nm_2300nm_0_planck.gif

Achtung: Variablenname vierer wird mit gespeichert

Temperature function saved as
wrfile = tempfkt_1950nm_2300nm_27x4.bin
with matrix-format vierer=[grad,e1,e2,e1zue2]

... GIF-Plot of Window 1 saved as
tempfkt_1950nm_2300nm_1.gif
... GIF-Plot of Window 2 saved as
tempfkt_1950nm_2300nm_2_r-b.gif
... GIF-Plot of Window 3 saved as
tempfkt_1950nm_2300nm_3_quot.gif

Temperature function saved under
wrfile = tempfkt_1950nm_2300nm_27x4.bin
with matrix-format vierer=[grad,e1,e2,e1zue2]

Name                Type         Size         Bytes
vierer              constant     27 by 4      880

for test:
-->size(vierer)
-->vierer
-->whos -name vierer

Example for reload:
-->wrfile
-->clear 'vierer'
-->load(wrfile)
-->vierer

End.

GIF-Files saved at
C:\Daten\_Bibliothek\Bits_and_Bytes\Scilab\wav2gif_TU
```

# tempfunktion_entwickeln.sce – Scilab-Windows

**Scilab-Graphic(0…3):**

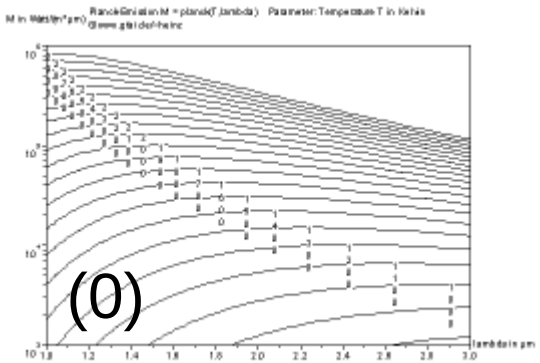
(0): Plancksche Schwarzkörperstrahlung

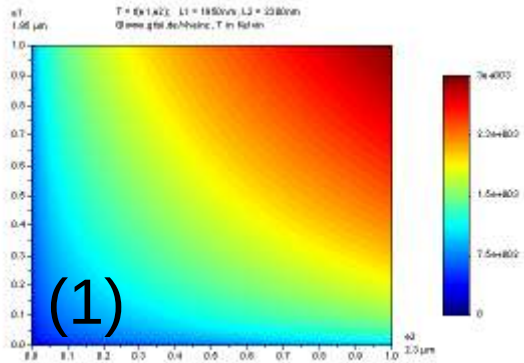
(1): 2d-Plot T(e1,e2)

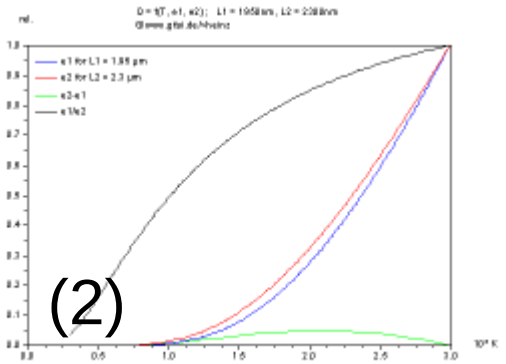
(2): Test-Funktionen

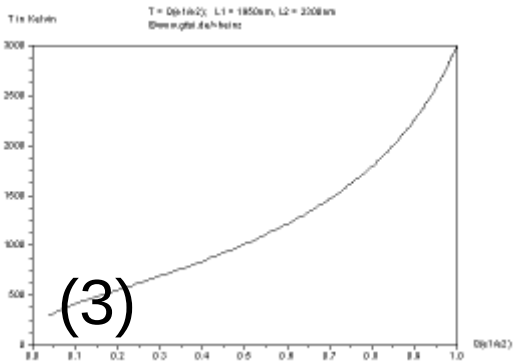
(3): Ausgabe-File

T = T(e1/e2)



# tempfunktion_plotten.sce

Input: *.bin Temperaturfunktion

Output: *.gif

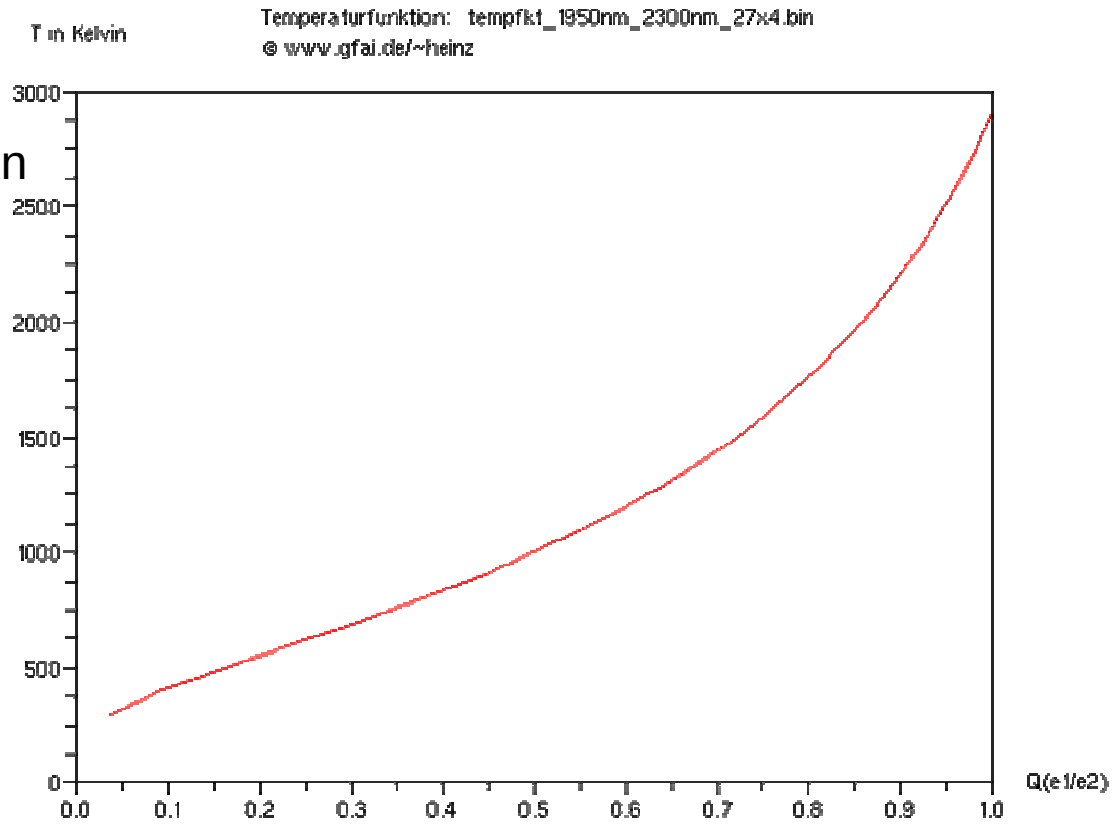
Die Funktion kann zum Einlesen der Temp.-kurve genutzt werden.

Einmal starten, dann z.B. auf Scilab-Kommandozeile eingeben:

→ who

→ wrfile

→ vierer



# tempfunktion_plotten.sce – Scilab-Dialog

```
Temperaturfunktion *.bin plotten
heinz@gfai.de

vierer = Matrix aus vier Spalten

Current working directory:
C:\Daten\_Bibliothek\Bits_and_Bytes\Scilab\wav2gif_TU

Input-File:
tempfkt_1950nm_2300nm_27x4.bin

Read-size: vierer in 27 Reihen mit 4 Spalten

vierer:
 1.Spalte: Temperatur in Kelvin
 2.Spalte: Werte e1
 3.Spalte: Werte e2
 4.Spalte: Quotient e1/e2

String-Matrix mit 27x4 Werten wurde als vierer(grad,e1,e2,ratio) gelesen

Tests:
-->vierer
-->size(vierer)
-->typeof(vierer)
```

# wav2gif_v5.sce – Wavs als Gifs plotten

Zum Plot von Zeitfunktionen und deren Verknüpfungen

Input:

*.wav zu plottende Zeitfunktionen

*.pra Parameter-Initialisierung - in diesem File werden alle Funktionen definiert und editiert. Er wird im Dialog abgefragt.

Output:

*.gif Plotbild der Zeitfunktionen

Funktion:

- Zweiteilig:
  - Plot von max. 8 Wav-Zeitfunktionen f(…) und
  - Plot von max. 8 aus Wavs berechneten Zeitfunktionen g(…)

# wav2gif_v5.sce – Parameterfile *.pra

```
// A664-0398_2010_04.pra
// Steuerfile für WAV2GIF.SCE v5
//
// Beispiel zur Namensgebung:
// Ursprung:      mist.chl          (source: mist)
// abgeleitet:    mist_IR.wav, mist_UV.wav, mist_blau.wav etc.
// "blau" ist z.B. als parameter_1 "filename" referenziert
//
parameter='Schweißparameter unbekannt';
source='A664-0398'; // chl-Stammname hier eintragen
//
// Plot-Überschrift
//
titel='Planck-Temperatur TP in 10.000 Kelvin';
//
//
// Plotbereich der wav-Files
tstart=106;                    // Start in Millisekunden
tdelta=5.;                     // in Millisekunden
//
// Anzeigebereich Plotbild
yo=-.1;          // Anfang Wertebereich y
yr=1.2;          // Ende Wertebereich y
//
// Labels - Legendenplatzierung
xL=170;          // Label-Startposition x
yL=.5;           // Label-Startposition y
yk=.1;           // Labelabstand y
//
// Wertebereich xo,xr wird durch delta und fs erzeugt
// Samplerate fs kommt aus *.wav
//
```

```
// Temperaturkurve laden:
kurve='tempfkt_1950nm_2300nm_27x4.bin';
//
// WAVs plotten:
// Parameter: 1_wavname 2_farbe 3_funktion 4_funktion 5_plot?
// für fx stehen alle Funktionen der Form f(y) und f(y,const) bereit
// erster Name muß mit *.wav -Kürzel identisch sein
//
anzf=6;
                                // Anzahl zu plottender WAVs, max.8
f1='Isch black Amp(y,100) I(y,1e3) yes'; // Amp für I-Kalibrierung
f2='Usch gray Volt(y,100) U(y,100) yes'; // Volt für U-Kalibrierung
f3='440nm blue scale(y,1) norma(y) yes';        //
f4='740nm red scale(y,1) norma(y) yes';         //
f5='1950nm purple scale(y,1) norma(y) yes';     //
f6='2300nm yellow scale(y,1) norma(y) yes';     //
//
//
// Funktionen berechnen:
// Parameter: 1_funktion 2_farbe 3_skalierung 4_plot?
// für Parameter 1, 3 und 4 stehen alle Funktionen bereit
// Zugriff auf f1...f6 möglich
//
anzg=1;
                                // Anzahl zu berechnender Zeitfkt., max.8
g1='TP(f5,f6,1) green scale(y,2e-4) yes';       // Planck-Temperatur
//g1='TP(f3,f4,1) green scale(y,1e-4) yes';     // Planck-Temperatur
//g3='P(f3,f4) orange scale(y,1e-4) yes'; // Power (scale in kW~10e-3, W~1)
//g4='E(f3,f4) orange scale(y,1) yes'; // zugef. Energy in mW pro Sample
//g5='S(f1,f2,0,-.1,1,0) purple scale(y,1) -'; // Schwellwert
//g6='f1-f2 yellow scale(y,1) norma(y) -'; // Subtraktion
//g7='f6-f5 green scale(y,1) yes';  // Differenz
```

# wav2gif_v5.sce – Scilab-Dialog

```
WAV-Plotprogramm WAVs to GIF
    Version v5 2009-10-20
    For research and education only
    © Copyrights: www.gfai.de/~heinz

    Programm kann in Scilab-Directory stehen
    Parameterfile *.PRA steuert Ausführung
    Temperaturfunktion *.BIN ist nutzbar
    *.PRA *.BIN und *.WAV sollten im gleichen Verzeichnis
       stehen
    Ablauf: Teil 1) WAVs einlesen und plotten
            Teil 2) Funktionen berechnen und plotten

Parameter-File A664-0398_2010_04.pra wird geladen
aus dem Verzeichnis:
 C:\Daten\_Bibliothek\Bits_and_Bytes\Scilab\wav2gif_TU

Arbeitsverzeichnis ist:
 C:\Daten\_Bibliothek\Bits_and_Bytes\Scilab\wav2gif_TU

Temperaturkurve wird geladen:
-->tempfkt_1950nm_2300nm_27x4.bin

b.w.
```

```
============ WAVs f lesen und plotten ==============

Aufgabe aus Parameter-File lesen:
f1 = Isch black Amp(y,100) I(y,1e3) yes
   String-Zerlegung:
   f1 ~ Isch
   black
   Amp(y,100)
   I(y,1e3)
   Plot? yes
   laden: A664-0398_Isch.wav
   1-te Zeitfunktion gelesen mit Sample-Rate 44100 Hz, 16 Bit/Sample
   Plot - Label xleg = 170, Label yleg = 0.5 0

Aufgabe aus Parameter-File lesen:
f2 = Usch gray Volt(y,100) U(y,100) yes
   String-Zerlegung:
   f2 ~ Usch
   gray
   Volt(y,100)
   U(y,100)
   Plot? yes
   laden: A664-0398_Usch.wav
   2-te Zeitfunktion gelesen mit Sample-Rate 44100 Hz, 16 Bit/Sample
   Plot - Label xleg = 170, Label yleg = 0.6 1


... ...
... b.w.
```

# wav2gif_v5.sce – Scilab-Dialog

```
...
...
====== Funktionen g berechnen und plotten ========

Aufgabe aus Parameter-File lesen:
g1=TP(f5,f6,1) green scale(y,2e-4) yes
   Berechnung von
    g1 = TP(f5,f6,1)
   Ergebnis ist 1 lang
Plots:  Label xleg = 170, Label yleg = 1.1

================ Plot beschriften =============

   Isch black Amp(y,100) I(y,1e3) yes
   Usch gray Volt(y,100) U(y,100) yes
   440nm blue scale(y,1) norma(y) yes
   740nm red scale(y,1) norma(y) yes
   1950nm purple scale(y,1) norma(y) yes
   2300nm yellow scale(y,1) norma(y) yes
   TP(f3,f4,1) green scale(y,1e-4) yes

Bild wurde gespeichert als File:
A664-0398_f42_g8.gif

Verzeichnis:
C:\Daten\_Bibliothek\Bits_and_Bytes\Scilab\wav2gif_TU
```

Ergebnisplot: