NASA/TM–20230002218

Numerical Study of Solidification Crack Susceptibility in Novel Refractory Alloy Systems

F.N. Michael and J.W. Sowards
Marshall Space Flight Center, Huntsville, Alabama

February 2023

The NASA STI Program…in Profile

The NASA STI Program collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NTRS Registered and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA's counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and mission, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

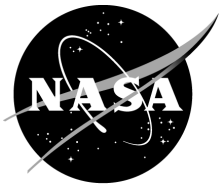
For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at http://www.sti.nasa.gov

- Help desk contact information:

https://www.sti.nasa.gov/sti-contact-form/ and select the "General" help request type.

NASA/TM–20230002218



Numerical Study of Solidification Crack Susceptibility in Novel Refractory Alloy Systems

F.N. Michael and J.W. Sowards
Marshall Space Flight Center, Huntsville, Alabama

National Aeronautics and
Space Administration

Marshall Space Flight Center • Huntsville, Alabama 35812

February 2023

TRADEMARKS

Available from:

NASA STI Information Desk
Mail Stop 148
NASA Langley Research Center
Hampton, VA 23681–2199, USA
757–864–9658

This report is also available in electronic form at
<http://www.sti.nasa.gov>

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS, SYMBOLS, AND ABBREVIATIONS

C	carbon
CALPHAD	Calculation of Phase Diagrams
CSSI	crack solidification susceptibility index
DBTT	ductile-brittle transition temperature
HEA	high entropy alloys
Hf	hafnium
HPC	high performance computing
ICME	integrated computational materials engineering
K	Kelvin
O	oxygen
Mo	molybdenum
N	nitrogen
Nb	niobium
Re	rhenium
SNP	Space Nuclear Propulsion
Ta	tantalum
TC	Thermo-Calc
TDB	thermodynamic database
TM	technical memorandum
V	vanadium
W	tungsten
wt.%	mass percent
Zr	zirconium

NOMENCLATURE

d	derivative
f_s	fraction of solid
T	temperature

TECHNICAL MEMORANDUM

NUMERICAL STUDY OF SOLIDIFICATION CRACK SUSCEPTIBILITY IN NOVEL REFRACTORY ALLOY SYSTEMS

1. INTRODUCTION

Many types of alloys are susceptible to cracking upon solidification.[1–5] For various alloys, cooling through the range of temperatures from liquidus to solidus, also known as the solidification range, presents many challenges in the production of crack-free structures. This is true in melt pool fusion welding and also in melt pool-based additive powder bed fusion, and even blown-powder and wire-fed 3D printing. While not the focus of this paper, these challenges are further exacerbated by cooldown to room temperature or even into cryogenic temperature ranges; cooling through the ductile-brittle transition temperature (DBTT) for the particular alloy's mass percent (wt.%) composition will result in exceeding the embrittlement limit, which will result in the development of cracks in the cast or manufactured article.[6,7]

These susceptibilities are often more pronounced in the case of refractory metals and their alloys. Some of the highest melt temperatures of materials are required for a melt and homogenization of a liquid solution consisting of a refractory metal alloy composition. Reaching homogenization results in very high liquidus and solidus temperatures; potentially results in large solidification ranges at these elevated temperatures; and also results in very high gradients of solidification temperatures. These high gradients near the solidus temperature, or near the full solidification fraction, are indicators of susceptibility to cracking. Past the solidus temperature, and continuing the cooldown, the DBTTs for refractory metals and their alloys are often considerably higher than room temperature. Cooldown past solidification to reach room temperature (or cooler) has to additionally pass through this embrittlement phase, necessitating additional considerations in processing to achieve crack-free casts and builds. It is apparent, then, that controlling both processing and composition are key to creating quality 3D additive prints, both for refractory alloys and other types of alloys.

It is, therefore, beneficial to be able to control elemental compositions of refractory alloys; for example, when the varied effects leading to solidification cracking are suppressed during a cast or a manufacturing weld, or during an additive melt resolidification process. The composition control approach, in which the wt.% composition chemistry is additionally chosen to enhance some aspects of the alloys' properties (e.g., phases' chemistries, resultant strengths, and ductility) while suppressing others (e.g., susceptibility to solidification cracking), is additionally an approach toward optimizing post-solidification alloy properties. In this technical memorandum (TM), the researchers will focus primarily on the solidification cracking issues in the transition from liquidus to solidus, and will leave processing controls and optimization methods for future work.

2. PROCEDURE

Recently, Calculation of Phase Diagrams (CALPHAD)[8]-based solidification computations, such as Scheil or equilibrium computations, have been utilized to propose crack solidification susceptibility indices (CSSIs) of cracking. These CSSIs have been proposed in order to predict the cracking susceptibility of an alloy in the solidification range, as a function of its solidification fraction and as dependent upon its wt.% elemental composition through the CALPHAD computation. Therefore, a metric is now available for the prediction, and presumably the control, of solidification cracking.

A CSSI index deemed to well represent the cracking susceptibility was recently proposed by Kou,[9,10] building on the work of Clyne and Davies.[11] This calculation obtains the maximum of the absolute value of the derivative of the solidification temperature (T) vs. the square root of the fraction of solid (f_s) curve obtained via CALPHAD thermodynamics-based (i.e., Scheil-type) solidification curves. The calculation is shown in equation 1.

$$\text{CSSI} = \max\left(\left|dT/df_s^{1/2}\right|\right) \tag{1}$$

There are several approximations employed by these solidification calculations, including equilibrium where an infinite time is assumed at each instance to equilibrate the process, and even solidification calculations where a quasi-equilibrium is achieved; and additionally where elements from the remaining liquid are allowed to (back-) diffuse into the solidified fraction. Variations include non-equilibrium solidification, in which a cooldown rate is required to perform the calculation, allowing for process control modeling in the casting, weld, or additive build. The researchers will apply the 'classic' Scheil solidification for the gradient approach where atomic diffusion is assumed to be zero in the solid phase and infinite in the liquid phase.

The typical reported gradient approach has been applied successfully to the solidification of lightweight alloys (e.g., aluminum alloys) as shown in figure 1.[10] The gradient approach has been discussed as the maximum gradient near the solidification point (solidus) to within 0.95 to 0.99 f_s.[9–14]

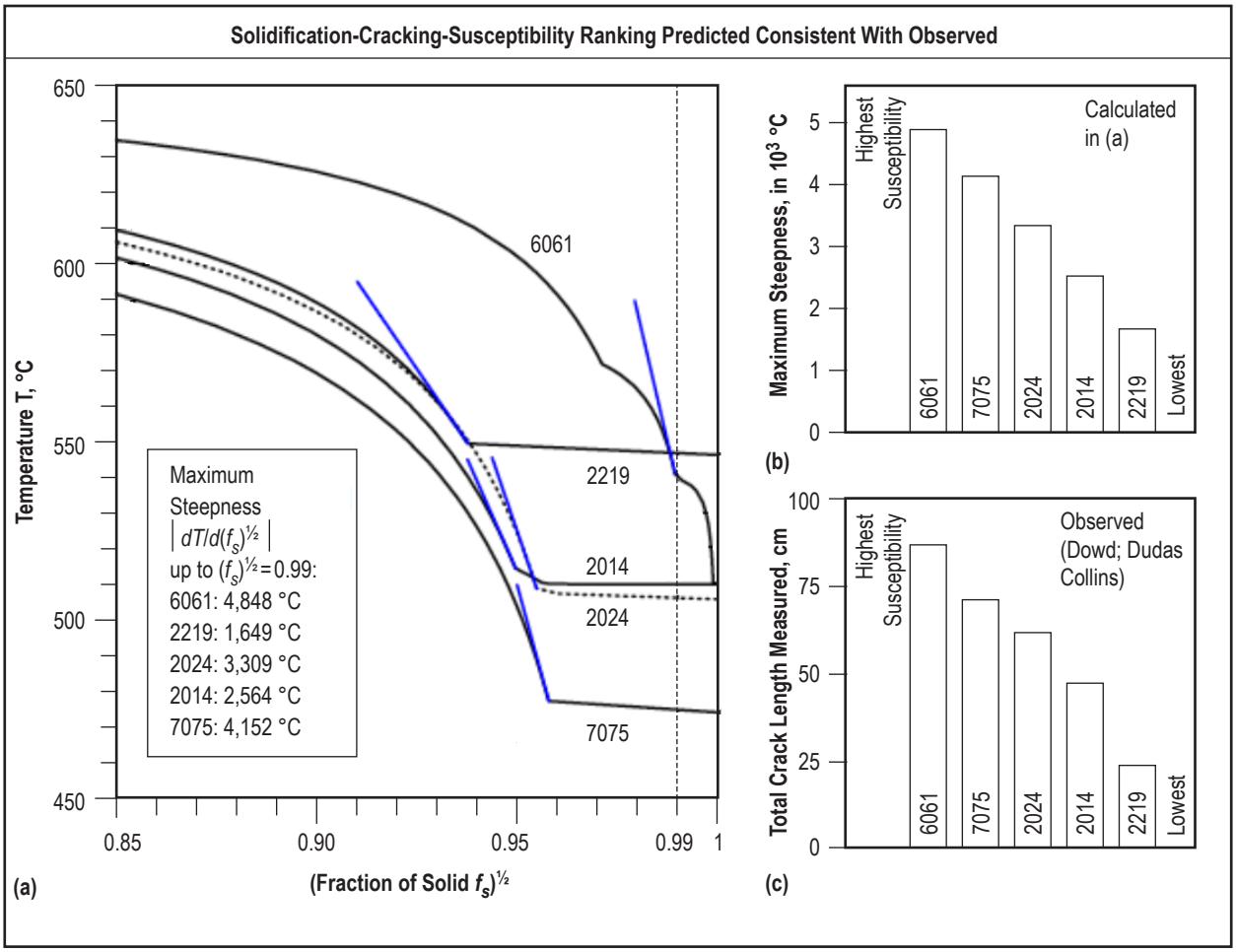


Figure 1. Kou type of near solidus (>0.95 $f_s^{½}$) maximum slope (i.e., gradient) of the solidification curve for several aluminum alloys.[10]

In this TM, the researchers apply this Kou gradient method to refractory alloys for the first time. This TM focuses on refractory alloys modeling, as NASA's aerospace mission is increasingly seeking to rely upon these high thermal performance alloys for its hypersonic and supersonic aerospace applications; its propulsion (e.g., engine nozzles, liners, etc.) and re-entry environments; and its Space Nuclear Propulsion (SNP) programs. For example, SNP relies on refractory alloys in order to build the design requirements of ≈1 m scale fuel elements comprised of a refractory alloy metal matrix interspersed with fuel and penetrated by coolant channels. These channels allow the flow of fuel (e.g., hydrogen), which then is heated to ≈3,000 K and allowed to expand and be exhausted in the aft direction, therefore providing thrust to the spacecraft. Additionally, the various nuclear fuels being investigated, which are to be homogeneously embedded into the built fuel element refractory alloy metal matrix, are either partially composed of refractory elements or are refractory metal- or alloy-clad, forming spherical fissile fuel pellets; and, as mentioned, may even comprise the fissile fuel composite itself.

The researchers also state that application of what has been termed the 'Kou-Scheil' approach to refractory metals and their alloys is warranted, based on a comparative analysis of refractory alloys CSSI and experimental data based on Varestraint cracking tests. The comparison is made in figure 2 via a Spearman rank correlation, which captures the normalized weldability ranking derived from the Varestraint test data, and the solidification CSSI in arbitrary units. Note a >0.80 correlation between model and experiment. This indicates that the model presented herein reasonably describes solidification cracking that has been measured empirically.

Table 1. Chemical composition of alloys shown in figure 2.

Alloy	Nominal Composition	Tantalum (Ta)	Niobium (Nb)	Tungsten (W)	Hafnium (Hf)	Molybdenum (Mo)	Rhenium (Re)	Vanadium (V)	Zirconium (Zr)	C ppm	O ppm	N ppm	C ppm	O ppm	N ppm
T-111	Ta-8W-2Hf	Balance	–	8.2	2	–	–	–	–	40	80	12	33	40	12
ASTAR-811C	Ta-8W-1Re-0.7Hf-0.0.025C	Balance	–	8.1	0.9	–	1.4	–	–	300	70	10	210	5	5
FS-85	Nb-27Ta-10W-1Zr	28.1	Balance	10.6	–	–	–	–	0.94	20	90	60	32	53	47
T222	Ta-9.6W-2.4Hf-0.01C	Balance	–	9.2	2.55	–	–	–	–	115	50	20	119	17	11
B-66	Nb-5Mo-5V-1Zr	–	Balance	–	–	5.17	–	4.89	1	95	110	63	37	120	70
Ta-10W	Ta-10W	Balance	–	9.9	–	–	–	–	–	50	40	20	5	10	10
SCb-291	Nb-10W-10Ta	9.83	Balance	10	–	–	–	–	–	20	110	40	22	101	20

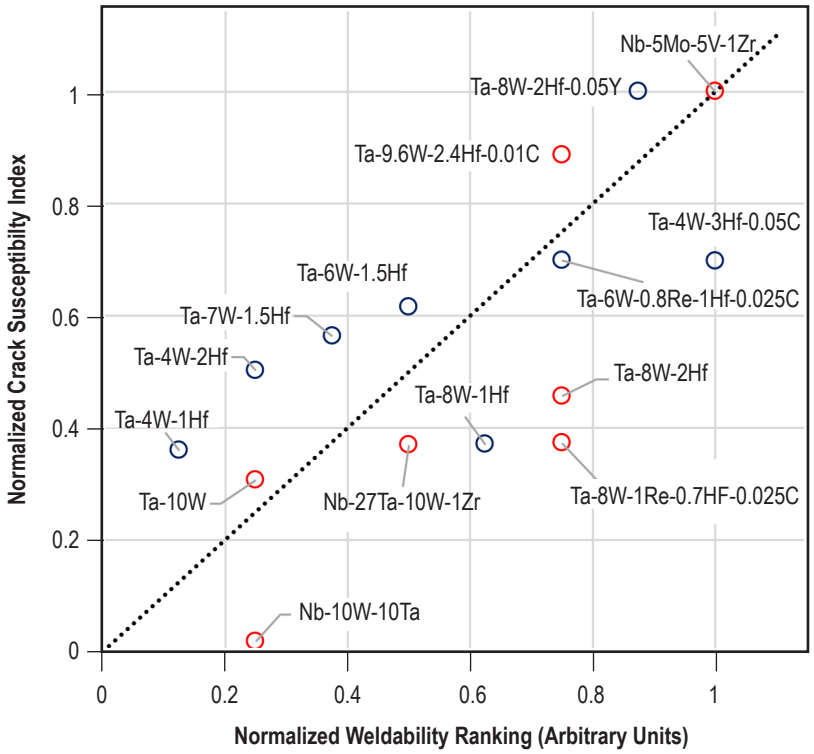


Figure 2. Model performance Spearman rank correlation in arbitrary units for several refractory alloys of interest.

In section 3 of this TM, the researchers therefore apply a Kou-Scheil gradient-based CSSI approach to several refractory alloys from within a Python-based module that calls on the Scheil solidification module of pycalphad[15,16] and the THERMO-CALC® (TC) Python interface module.[17] The module's code is provided in the appendix. The limitations and challenges of the computational module and approach will be discussed to some degree. Therefore, the researchers suggest some refractory alloys compositions that may guide experimental investigations; for instance, via alloy synthesis and characterization, and subsequent weldability and printability process development and testing. These investigations form a specific set of instances of the integrated computational materials engineering (ICME) workflow and overarching paradigm.[18]

3. DISCUSSION

A refractory metal of high interest is tungsten (W), which has the highest known melting temperature. Tungsten forms the foundation of a large portion of experimental and development efforts at NASA and other aerospace-focused parties in industry and academia. Therefore, this technical memorandum will have an outsized focus on W-based alloys in CALPHAD CSSI modeling.

When melted and while undergoing solidification, W experiences a susceptibility to cracking at the near-fully solidified region of $f_s^{1/2}$ near the 0.95–0.99 range. This is identified as the region where liquid fraction filling of dendritic growth spacing is unbalanced in relation to the unfilled regions, which (in combination with thermal shrinkage or expansion and contraction (i.e., stresses and strains)) results in pre-full solidification cracking.

Therefore, three strategies are available in controlling and suppressing this solidification cracking susceptibility:

(1) Alloying via addition of useful elements in order to suppress or enhance various solidification dynamics, such as nucleation dynamics, grain growth dynamics (i.e., pinning), eutectic and intermetallic phases, the solidification range itself, etc.

(2) Control of the cooldown solidification temperatures and, therefore, the environment of the solidification process.

(3) Nanoparticle and/or dispersoid additions for grain refinement and equiaxed grain production, as well as in-situ nanometer scale metal carbide and/or metal oxide formation.

(4) A hybrid or combination of the previous approaches.

This TM focuses on the first approach, that of alloying. Additionally, the Scheil solidification approach utilized in this research does not include diffusion into the solidified fraction, although that is easily implementable in the TC version discussed in this memorandum, in addition to the pycalphad approach. The researchers utilized the open-source Scheil 0.8.5 and the pycalphad module version 0.1.2,[16] as well as the TC Python interface.[17] All Python module runs were performed on the Jupyter Lab environment. In the case of pycalphad, the COST507 thermodynamic database (TDB) was utilized (as well as others). Any future user of the code presented in this TM may use any standard metal thermodynamic database with the TDB format to model their specific metallurgical system. There are numerous open-source TDB available at online repositories. For example see the Thermodynamic Database Database maintained at Brown University. (https://avdwgroup.engin.brown.edu/).

The researchers have written a Python-based module for calculation of the Kou-Scheil CSSI, which is presented in the appendix. The python module is applicable directly to the open-source pycalphad python-based CALPHAD software available on GitHub. It is also useful for the Thermocalc TC software as a python based scripting module with minor modifications of the Scheil calls, though this commercial python add-on modification is not shown. The computations proceeded rapidly on a high-performance computing (HPC) workstation for binary, ternary, quaternary, and quinary refractory alloys. Note that quaternary and quinary often required extended computational times even on a 32-core HPC desktop workstation. One reason for the increased computational time is that adding additional calculation density of points for the adaptive calculation(s) increases the calculation time significantly. Multiple phases and rapid deviations, if these occur in the wt.% space modeled in a quaternary or quinary alloy, require this adaptive approach and a higher number (i.e., density) of points.

3.1 Refractory Alloy Examples

A ternary W-Re-Ta refractory alloy was examined via the Kou-Scheil gradient CSSI method and a gradient 'heat' map was generated to show the change in solidification cracking susceptibility as the wt.% of Re and Ta was changed (fig. 3). The units of the gradient are in degrees Kelvin, yet it must be noted that the gradient is useful as an index relative to itself. The information contained in the temperature gradient is also useful for other considerations, such as the thermal expansion near the solidus point; yet as a cracking index, its real value lies in the ability to determine regions of high and low vulnerability to cracking. The researchers performed CSSI calculations in the COST507 TDB and the pycalphad module.

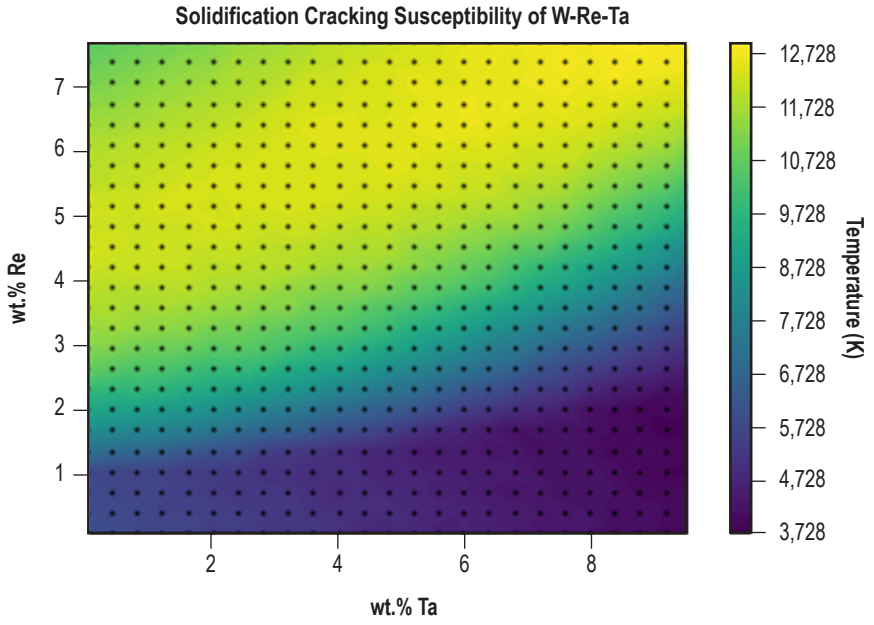


Figure 3. Ternary solidification cracking susceptibility map made via Kou-Scheil gradient approach for W-Re-Ta. The colored 'heat' map shows regions of low cracking susceptibility, which are predominantly in low wt.% Re content.

The method can be compared to the time spent in the vulnerable region utilized by the alternative TC approach to CSSI crack solidification index determination. The calculations relied upon TC's Nickels database. The researchers performed this analysis, shown in figure 4, and obtained good agreement with the Kou-Scheil gradient. It was immediately apparent the two approaches yielded good agreement for the ternary refractory alloy being considered.

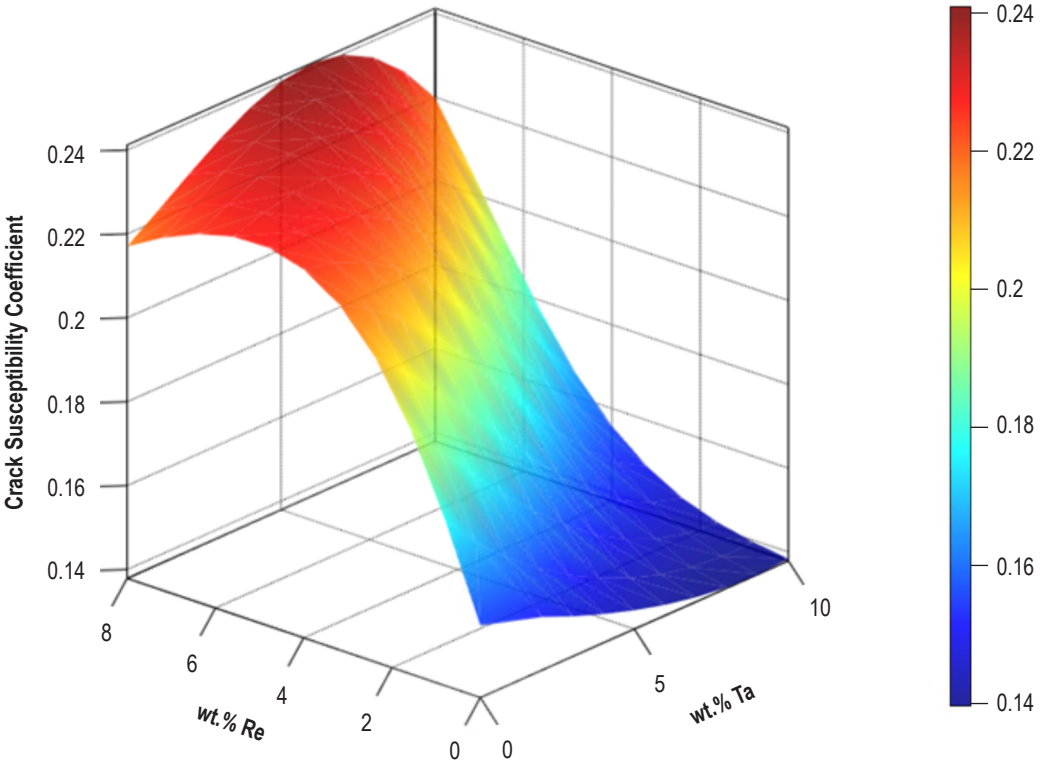


Figure 4. Thermo-Calc (TC) W-Re-Ta CSSI crack solidification index determination based on the time spent in the vulnerable region during solidification.

The researchers also performed similar calculations for additional alloys in the COST507 TDB and the pycalphad module. Figures 5 and 6 show the results of investigating a W-Re-Ta-C system from both approaches, relying on COST507 and the HEA TDB in TC. Again, the results are in agreement. The peak CSSI in both figures indicates similarly in both approaches that, in terms of wt.% C, the alloy's most vulnerable CSSI region is near 0.05 wt.% C.

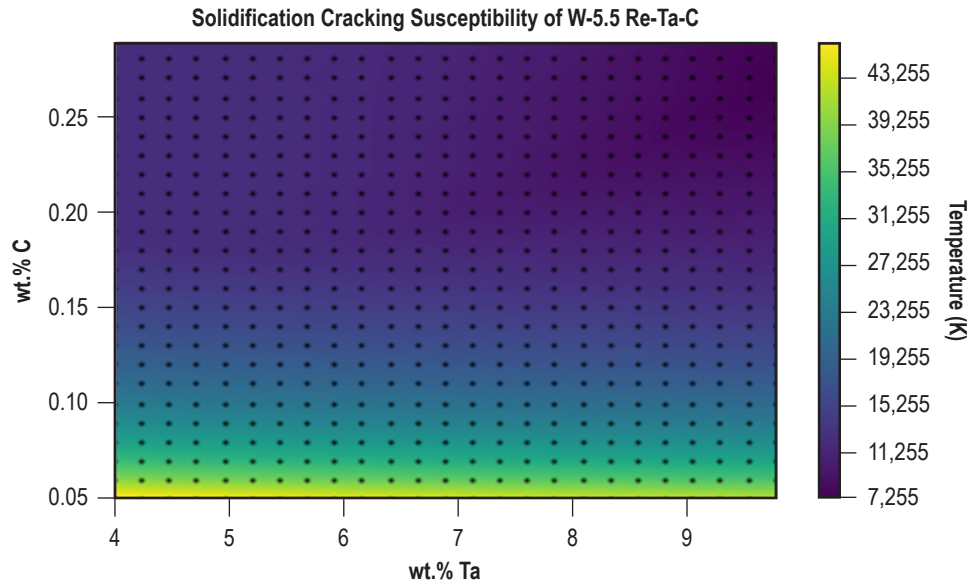


Figure 5. Kou-Scheil CSSI calculation in mass percent, with a maximum of 10 wt.% Ta and 0.3 wt.% C content. Rhenium (Re) is fixed at 5.5 wt.% with W making up the remainder.

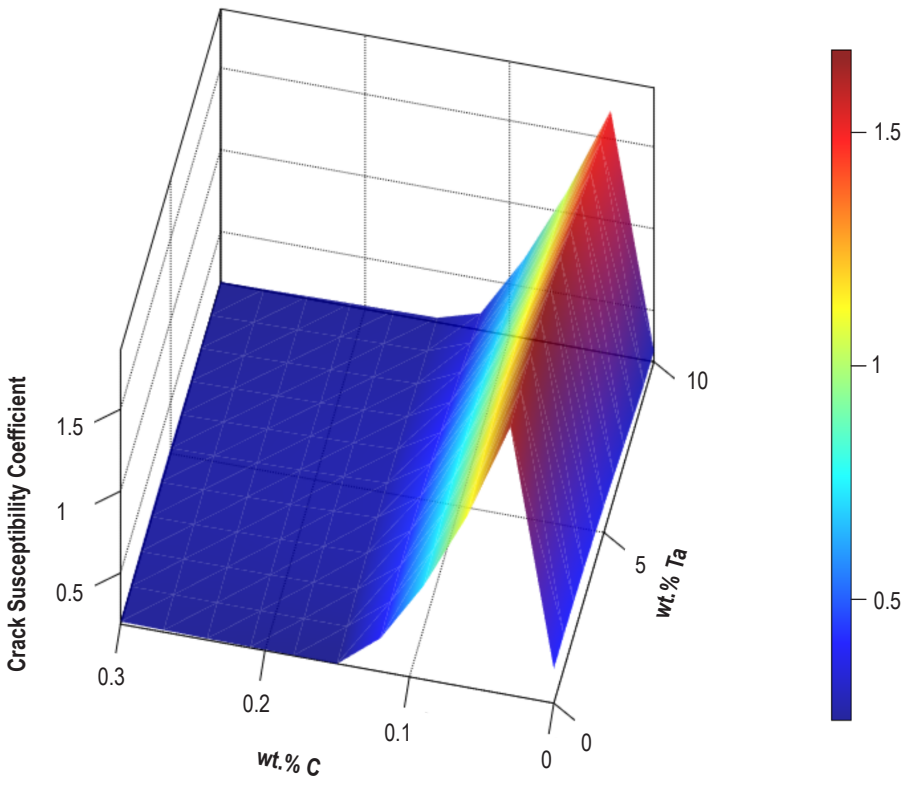


Figure 6. The TC analog of figure 5's CSSI via the time spent in the vulnerable region approach in TC.

4. CONCLUSIONS

Whether the melt pool is the result of an additive manufacturing 3D print or of a welding or joining process—despite these techniques having significantly different boundary conditions and even process-dependent sensitivities—the melt pool chemistry needs to be analyzed as a factor in and of itself in order to contribute to the successful build of a crack-free part. The Kou-Scheil gradient approach, which was previously applied to lightweight alloys, has been applied in this research to refractory high-temperature alloys relevant to NASA's aerospace mission. The researchers presented two examples of the application of the method and its results, and compared the results to existing and alternative approaches to determining susceptibility to cracking, as by TC's vulnerability time ratio with regard to non-vulnerability time approach. The researchers found the two approaches to be in good agreement. Additionally, it was shown via a Spearman rank that the CSSI approach correlated well with experimental data for refractory alloys.

The researchers, therefore, conclude that the various crack susceptibility approaches are complementary and are applicable to refractory alloy cracking mitigation, which is necessary for the various refractory alloys-based NASA missions. Furthermore, the Kou-Scheil approach is expected to be applicable as a calculation method to recent research efforts at NASA and with partners, which seek to include pressure head and geometrical interdendritic spacing features into a materials-agnostic approach for a further refined CSSI.

The researchers make available the Python-based script and modules as an appendix to this TM. This approach is suggested as an open-source alternative or complement to solidification cracking predictions using commercial software.

APPENDIX—PYTHON-BASED MODULE FOR CALCULATION OF THE KOU-SCHEIL CRACK SOLIDIFICATION SUSCEPTIBILITY INDEX

KOU-SCHEIL MODULE V.1.4
FEBRUARY 2, 2023

[ ]:
```
#Import dependencies
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
from pycalphad import Database, binplot, equilibrium, variables as v
from scheil import simulate_scheil_solidification
import time
import math
import scipy.signal
import pandas as pd
plt.rcParams.update({'font.family':'arial'})
```

[ ]:
```
#Initial full phases caculation. Run a test example & ensure phases & ranges␣
 ↪are being determined properly.
# Note: you can use any compatible name.tdb database\
# The inputs are in wt%...which are converted to molar fractions via the␣
 ↪get_mole-fraction() utility.
# Start temperature should be above the melt liquid temperature
# Check points can be uncommented if checks are needed

# First calculate a scheil plot with desired database and nominal chemistry for␣
 ↪visual check for expected behavior
# Setup the simulation parameters that will carry to sweep

#Define Elements.  Additional terms may be added although only two are varied.
a = 'W'  #Use the "solvent" metal here, variable is a string.
b = 'TA'  #Use the first "solute" metal here, variable is a string.
c = 'C'  #Use the second "solute" metal here, variable is a string.

#Define Database
dbf = Database(open('COST507-modified.tdb', encoding='latin-1').read())  ␣
 ↪#cost507 must be the pycalphad compatible version
#dbf = Database(open('NIST-NiMob13.tdb', encoding='latin-1').read())   #example␣
 ↪of alternative TDB
comps = [a, b, c, 'VA']
phases = sorted(dbf.phases.keys())
print('phases =    ',phases)
```

```
#Define Simulation Conditions
start_temperature = 3800     # [Kelvin]
dT= 5.0                      # Temperature step size [Kelvin]
Filter = 0.90                # Defines a sqrt(fs) value above which max value is␣
 ↪obtained
Stop = 0.01                  # Simulation ends when this fraction of liquid is␣
 ↪achieved
start_time = time.time()     # Setup clock to track computation time

#this mass fraction approach uses weight percent from mass fraction and␣
 ↪converts it to mole fraction as input to Scheil module
mass_fractions={v.W(b): 0.1, v.W(c): 0.1}    # Input composition of b and c.␣
 ↪The command is v.W('element') for mass fraction wt%
print("Mass fractions =",mass_fractions)       # check point
initial_composition_pre = v.get_mole_fractions(mass_fractions,a, dbf)  # this␣
 ↪then obtains/calcs the molar fraction from the input wt%
dd = initial_composition_pre
# print(' dd values=','  ',dd)   # check point to confirm {b, c, ...} are␣
 ↪obtained
key1, value1 = list(dd.items())[0]   # extract the vaules, as value1, value2, ..
 ↪.these are now molar fractions
key2, value2 = list(dd.items())[1]
initial_composition = {v.X(b): np.round(value1,6), v.X(c): np.round(value2,6)}␣
 ↪#input the from wt%, the molar fracs into the initial_composition
print("Mol fractions =",initial_composition)  # check point to confirm wt.␣
 ↪fraction computed correctly

# perform the Scheil simulation
sol_res = simulate_scheil_solidification(dbf, comps, phases,␣
 ↪initial_composition, start_temperature, step_temperature=dT, verbose=False,
                                          eq_kwargs={'calc_opts': {'pdens':␣
 ↪1000}}, adaptive=True, stop=Stop)
fsnew = np.sqrt(sol_res.fraction_solid)
Tnew = np.abs(sol_res.temperatures)

# values like xvals' 4000-5000 represents number of points to interpolate/make␣
 ↪between T.min and T.max
# savgol is the Savistsky-Golay smoothing. Useful for solidification paths that␣
 ↪are 'discontinuous', meaning
# susceptible to computational artifacts.
xnew = fsnew #reassign dummy vars
xvals=np.linspace(0, 1, 4000)  #declare spacing and number of interpolation␣
 ↪points
power_smooth = np.interp(xvals, xnew, Tnew)   #apply interpolation
```

```
power_smooth_savitsky= scipy.signal.savgol_filter(power_smooth, 31, 3,␣
 ↪mode='nearest') #apply savitsky-golay smoothing/filter
power_smooth= power_smooth_savitsky  #reassign dummy vars
derivative = np.abs(np.gradient(power_smooth) / np.gradient(xvals))   #apply␣
 ↪grad method
xvalsn=xvals # np.delete(xvals, 1, 0)

#The following code plots three diagrams useful for interpretation of the␣
 ↪results.
plt.plot(sol_res.fraction_solid, sol_res.temperatures, label='FS')
plt.xlabel('Fraction solid, $f_s$', fontsize=14)
plt.ylabel('Temperature (K)', fontsize=14)
plt.title(str(a)+'-'+str(100*np.round(value1,4))+str(b)+'-'+str(100*np.
 ↪round(value2,4))+str(c)+'(at.%) Scheil Simulation', fontsize=14)
plt.xlim(0.0, 1.)
plt.show()
plt.plot(xnew,Tnew)     # plot the scheil interpolated curve and overlayed with␣
 ↪scatter (red)
plt.scatter(xvals,power_smooth, c='Red', s=1)
plt.xlabel(r'$\sqrt{f_s}$', fontsize=14)
plt.ylabel('Temperature (K)', fontsize=14)
plt.title(str(a)+'-'+str(100*np.round(value1,4))+str(b)+'-'+str(100*np.
 ↪round(value2,4))+str(c)+' (at.%) Scheil Simulation', fontsize=14)
plt.xlim(0., 1.)
plt.show()
plt.plot(xvalsn,derivative)
plt.xlabel(r'$\sqrt{f_s}$', fontsize=14)
plt.ylabel(r'$dT / df^\frac{1}{2}_s$', fontsize=14)
plt.title(str(a)+'-'+str(100*np.round(value1,4))+str(b)+'-'+str(100*np.
 ↪round(value2,4))+str(c)+' (at.%) Solidification Cracking Index',
          fontsize=14)
plt.xlim(0., 1.)
plt.show()

index = np.where(xvalsn>=Filter)  # a highpass filter that removes crack␣
 ↪susceptibility values for all values below "Filter"
max_value_filter = np.max(derivative[index])
max_value_any =  max(derivative) # choose the maximum of the derivative/
 ↪gradient values
solidus = sol_res.temperatures[-1]
print("-------------------------------------------------------------------------
      ------------------------------------------------")
print("Total Run time = %s seconds" % round((time.time() - start_time),1))
print("Composition =",mass_fractions)
print("Max CSI =",round(max_value_any,1),"K,",
      "Max CSI with Filter =",round(max_value_filter,1),"K,",
```

```
      "Solidus Temperature =",round(sol_res.temperatures[-1],1),"K"
        )
print("-------------------------------------------------------------------------
      ------------------------------------------------------")
```

[ ]:
```
# The iterative Kou-Scheil calculation is performed here for X-Y-Z
# The pdens=#, adaptive=True is a Scheil module feature and is very useful for␣
 ↪more complex, phases dependent solidification.
# The interpolated points are to add regularly spaced points (many..>pdens) for␣
 ↪the derivative
#     or gradient apprach to function properly on. Avoiding discontinuity␣
 ↪artifacts.
# The smoothing (Savitsky-Golay) is useful to regularize the solidification␣
 ↪path,
#     also for best derivatives or gradients. Optional.
# It is recommended that the user uncheck the Derivative plotters in order to␣
 ↪scan through the computations
#     ...detrmination of proper selection of the maximum derivative values is␣
 ↪key.
#

liquid_phase_name = 'LIQUID'
xdata = []       #clear and define array for element a/x
ydata = []       #clear and define array for element a/y
CSI_all = []     #clear and define array for max CSI across range of sqrt(fs)
CSI_filter = []  #clear and define array for max CSI above "Filter" setpoint of␣
 ↪sqrt(fs)
theta = []       #clear and define array for non-equilibrium solidus temperature

#Setup Grid Spacing
n = 10            #Element 'a'/x point density
m = 10            #Element 'b'/y point density
sum = 0          #Counter initilization
x = np.linspace(0.001, 0.05, num=n, endpoint = True)    #optional, define␣
 ↪spacing for Element x (min, max, number of points)
y = np.linspace(0.001, 0.1, num=m, endpoint = True)    #optional, define␣
 ↪spacing for Element y (min, max, number of points)
#y = np.logspace(-5, -3, num=m, endpoint = True)       #optional, log space for␣
 ↪dilute solute, define spacing for Element y (min, max,
#number of points)
start_time = time.time()
for i in x:   #loop through Element 'a'/x concentration, i
    for j in y:   #loop through Element 'b'/y concentration, j for each i
        #this mass fraction approach uses weight percent from mass fraction and␣
 ↪converts it to mole fraction as input to Scheil module
```

```
      mass_fractions={v.W(b): i, v.W(c): j}    # the command is v.
↪W('element') for mass fraction wt%
      initial_composition_pre = v.get_mole_fractions(mass_fractions,a, dbf) ␣
↪# this then obtains/calcs the molar fraction from the input wt%
      dd=initial_composition_pre
      key1, value1 = list(dd.items())[0]   # extract the vaules, as value1,␣
↪value2, ...these are now molar fractions
      key2, value2 = list(dd.items())[1]
      initial_composition = {v.X(b): np.round(value1,5), v.X(c): np.
↪round(value2,5)} #input the from wt%, the molar fracs into the␣
↪initial_composition
      # Regarding "pdens" kwarg below.  This is used to help convergence at␣
↪phase transformation points.  See Scheil module manual for explanation.
      sol_res = simulate_scheil_solidification(dbf, comps, phases,␣
↪initial_composition, start_temperature=start_temperature,␣
↪step_temperature=dT,
                                               verbose=False,␣
↪eq_kwargs={'calc_opts': {'pdens': 1000}}, adaptive=True, stop=Stop)
      fsnew = np.sqrt(sol_res.fraction_solid)
      Tnew=np.abs(sol_res.temperatures)
      # values like xvals' 4000-5000 represents number of points to␣
↪interpolate/make between T.min and T.max
      # savgol is the Savistsky-Golay smoothing. Useful for solidification␣
↪paths that are 'discontinuous', meaning
      # susceptible to computational artifacts.
      sol_res = simulate_scheil_solidification(dbf, comps, phases,␣
↪initial_composition, start_temperature=start_temperature,␣
↪step_temperature=dT,
                                               verbose=False,␣
↪eq_kwargs={'calc_opts': {'pdens': 1000}}, adaptive=True, stop=Stop)
      xnew = fsnew #reassign dummy vars
      xvals=np.linspace(0, 1, 4000)  #declare spacing and number of␣
↪interpolation points
      power_smooth = np.interp(xvals, xnew, Tnew)   #apply interpolation
      power_smooth_savitsky= scipy.signal.savgol_filter(power_smooth, 31, 3,␣
↪mode='nearest') #apply savitsky-golay smoothing/filter
      power_smooth= power_smooth_savitsky  #reassign dummy vars
      plt.plot(sol_res.fraction_solid, sol_res.temperatures, label='FS')
      plt.xlabel('Fraction solid, $f_s$', fontsize=14)
      plt.ylabel('Temperature (K)', fontsize=14)
      plt.title(str(a)+'-'+str(100*np.round(value1,4))+str(b)+'-'+str(100*np.
↪round(value2,4))+str(c)+'(at.%) Scheil Simulation', fontsize=14)
      plt.xlim(0.0, 1.)
      plt.show()
      plt.plot(xnew,Tnew)     # plot the scheil interpolated curve and␣
↪overlayed with scatter (red)
```

```
        plt.scatter(xvals,power_smooth, c='Red', s=1)
        plt.xlabel(r'$\sqrt{f_s}$', fontsize=14)
        plt.ylabel('Temperature (K)', fontsize=14)
        plt.title(str(a)+'-'+str(100*np.round(value1,4))+str(b)+'-'+str(100*np.
↪round(value2,4))+str(c)+' (at.%) Scheil Simulation', fontsize=14)
        plt.xlim(0., 1.)
        plt.show()
        derivative = np.abs(np.gradient(power_smooth) / np.gradient(xvals))  ␣
↪#apply grad method
        xvalsn=xvals # np.delete(xvals, 1, 0)
        plt.plot(xvalsn,derivative)
        plt.xlabel(r'$\sqrt{f_s}$', fontsize=14)
        plt.ylabel(r'$dT / df^\frac{1}{2}_s$', fontsize=14)
        plt.title(str(a)+'-'+str(100*np.round(value1,4))+str(b)+'-'+str(100*np.
↪round(value2,4))+str(c)+' (at.%) Solidification Cracking Index',
                  fontsize=14)
        plt.xlim(0., 1.)
        plt.show()
        index = np.where(xvalsn>=Filter)
        max_value_filter = np.max(derivative[index])
        max_value_any =  max(derivative) # choose the maximum of the derivative/
↪gradient values
        solidus = sol_res.temperatures[-1]
        sum=sum+1    # run counter
      ␣
↪print("-------------------------------------------------------------------------
              ------------------------------------------")
        print("Run # =  ", sum)
        print("Total Run time = %s seconds" % round((time.time() -␣
↪start_time),1))
        print("Composition =",mass_fractions)
        print("Max CSI =",round(max_value_any,1),"K,",
              "Max CSI with Filter =",round(max_value_filter,1),"K,",
              "Solidus Temperature =",round(sol_res.temperatures[-1],1),"K"
                )
      ␣
↪print("-------------------------------------------------------------------------
              ------------------------------------------")
        xdata += [i]
        ydata += [j]
        CSI_all += [max_value_any]
        CSI_filter += [max_value_filter]
        theta += [solidus]

#Export to csv:  'a'/x-->xdata, 'b'/j-->ydata.  This section is optional.  The␣
↪output is saved to pandas dataframes and csv files for subsequent
```

```
#plotting.
xdatapd = pd.DataFrame(data=xdata)
ydatapd = pd.DataFrame(data=ydata)
CSI_allpd = pd.DataFrame(data=CSI_all)
CSI_filterpd = pd.DataFrame(data=CSI_filter)
soliduspd = pd.DataFrame(data=theta)
xdatapd.to_csv('x.csv', index=False)
ydatapd.to_csv('y.csv', index=False)
CSI_allpd.to_csv('CSI_all.csv', index=False)
CSI_filterpd.to_csv('CSI_filter.csv', index=False)
soliduspd.to_csv('solidus.csv', index=False)
```

[ ]:
```
# Plot the X-Y-Z Kou_Scheil Crack Susceptobility 'Index' as a 2D contour plot␣
 ↪with Y-Z varied.
#plot filtered CSI data - minimimum value of fraction solid used as filter
zdatafilter = np.minimum(CSI_filter, min(CSI_filter))
f, ax = plt.subplots() #subplots(1, 2, sharex=True, sharey=True, figsize=(15,9))
cf1=ax.tricontourf(xdata,ydata,CSI_filter, np.arange(min(CSI_filter),␣
 ↪max(CSI_filter), 100)) # set the colorized indicator, and choose
#min/max and step-size
plt.colorbar(cf1,)
ax.plot(xdata,ydata, 'ko', markersize=1)
plt.yscale("log")
plt.ylabel(str(c)+' (Wt. fraction)')
plt.xlabel(str(b)+' (Wt. fraction)')
plt.title('Solidification Cracking Susceptibility of␣
 ↪'+str(a)+'-x'+str(b)+'-y'+str(c))
plt.savefig('test.png')
```

[ ]:
```
# Plot the X-Y-Z Kou_Scheil Crack Susceptobility 'Index' as a 2D contour plot␣
 ↪with Y-Z varied.
# Plot unfiltered CSI calculations
zdatafilter = np.minimum(CSI_all, min(CSI_all))
f, ax = plt.subplots() #subplots(1, 2, sharex=True, sharey=True, figsize=(15,9))
cf1=ax.tricontourf(xdata,ydata,CSI_all, np.arange(0, max(CSI_all), 100)) # set␣
 ↪the colorized indicator, and choose min/max and step-size
plt.colorbar(cf1,)
ax.plot(xdata,ydata, 'ko', markersize=1)
plt.yscale("log")
plt.ylabel(str(c)+' (Wt. fraction)')
plt.xlabel(str(b)+' (Wt. fraction)')
plt.title('Solidification Cracking Susceptibility of␣
 ↪'+str(a)+'-x'+str(b)+'-y'+str(c))
plt.savefig('test.png')
```

[ ]:
```

```

REFERENCES

1. Thomas, W.M.; Nicholas, E.D.; Needham, J.C.; et al.: U.S. Patent No. 5,460,317A, October 24, 1995.

1. Aucott, L.; Huang, D.; Dong, H.B.; et al.: "Initiation and Growth Kinetics of Solidification Cracking During Welding of Steel," *Scientific Reports*, Vol. 7, p. 40255, doi:10.1038/srep40255, January 11, 2017.

2. Zhang, F.; Liang, S.; Zhang, C.; et al.: "Prediction of Cracking Susceptibility of Commercial Aluminum Alloys during Solidification," *Metals*, Vol. 11, No. 9, p. 1479, doi:10.3390/met11091479, September 17, 2021.

3. Chen, S.; Ye, X.-X.; Tsang, D.K.L.; et al.: "Welding Solidification Cracking Susceptibility and Behavior of a Ni-28W-6Cr Alloy," *Journal of Materials Science & Technology*, Vol. 35, No. 1, pp. 29–35, doi:10.1016/j.jmst.2018.09.013, January, 2019.

4. Wang, C.; Yu, J.; Zhang, Y.; and Yu, Y.: "Phase Evolution and Solidification Cracking Sensibility in Laser Remelting Treatment of the Plasma-sprayed CrMnFeCoNi High Entropy Alloy Coating," *Materials & Design,* Volume 182, p. 108040, doi:10.1016/j.matdes.2019.108040, November 15, 2019.

5. Lukin, V.I.; Semenov, V.N.; Starova, L.L.; et al.: "Formation of Hot Cracks in Welding of Refractory Alloys," *Metal Science and Heat Treatment*, Vol. 43, pp. 476–480, doi:10.1023/A:1014896821621, November, 2001.

6. Yin, C.; Terentyev, D.; Pardoen, T.; et al.: "Ductile to Brittle Transition in ITER Specification Tungsten Assessed by Combined Fracture Toughness and Bending Tests Analysis," *Materials Science and Engineering: A*, Vol. 750, pp. 20–30, doi:10.1016/j.msea.2019.02.028, March 18, 2019.

7. Schwartzberg, F.R.; Ogden, H.R.; and Jafee, R.I.: "Ductile-Brittle Transition in the Refractory Metals," U.S. Department of Energy OSTI–4253559, 30 pp., 1959.

8. Spencer, P.J.: "A brief history of CALPHAD," *Calphad,* Vol. 32, No. 1, pp. 1–8, doi:10.1016/j.calphad.2007.10.001, March, 2008.

9. Kou, S.: "Solidification and Liquation Cracking Issues in Welding," *Journal of the Minerals, Metals, & Materials Society,* Vol. 55, No, 6, pp. 37–42, doi:10.1007/s11837-003-0137-4, June, 2003.

10. Kou, S.: “Predicting Susceptibility to Solidification Cracking and Liquation Cracking by CALPHAD,” *Metals,* Vol. 11, No. 9, p. 1442, doi:10.3390/met11091442, September, 2021.

11. Clyne, T.W.; and Davies, G.J.: “The Influence of Composition on Solidification Cracking Susceptibility in Binary Alloy Systems,” *The British Foundryman,* Vol. 74, No. 4, pp. 65–73, 1981.

12. Pakari, S.: “A criterion for cracking during solidification,” *Acta Materialia*, Vol. 88, No. 11, pp. 366–374, doi:10.1016/j.actamat.2015.01.034, April, 2015.

13. Liu, J.; and Kou, S.: “Crack susceptibility of binary aluminum alloys during solidification,” *Acta Materialia*, Vol. 110, pp. 84–94, doi:10.1016/j.actamat.2016.03.030, May 15, 2016.

14. Soysal, T.: “A criterion to find crack-resistant aluminium alloys to avoid solidification Cracking,” *Science and Technology of Welding and Joining*, Vol. 26, No. 2, pp. 99–105, doi:10.1080/13621718.2020.1843260, November 06, 2020.

15. Otis, R.; and Liu, Z.-K.: “pycalphad: CALPHAD-based Computational Thermodynamics in Python,” *Journal of Open Research Software*, Vol. 5, No. 1, doi:10.5334/jors.140, January, 2017.

16. Zenodo: “pycalphad-scheil,” V. 0.1.2, <https://zenodo.org/record/3630657#.Y3VGGvfMI2w>, Uploaded January 29, 2020.

17. Thermo-Calc Software AB: “TC-Python Documentation,” No. 2022b, < https://thermocalc.com/content/uploads/Documentation/Current_Static/tc-python-api-programmer-guide.pdf>, June 3, 2022.

18. Wang, W.Y.; Tang, B.; Lin, D.; et al.: “A Brief Review of Data-driven ICME or Intelligently Discovering Advanced Structural Metal Materials: Insight Into Atomic and Electronic Building Blocks,” *Journal of Materials Research,* Vol. 35, pp. 872–889, doi:10.1557/jmr.2020.43, April, 2020.

National Aeronautics and
Space Administration
IS63
George C. Marshall Space Flight Center
Huntsville, Alabama 35812

National Aeronautics and
Space Administration
IS63
George C. Marshall Space Flight Center
Huntsville, Alabama 35812