Mutiscale materials modeling: new takes on old problems

Maria Emelianenko

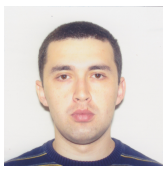
Dept. of Mathematical Sciences
George Mason University

IPAM Materials Defects Workshop, Nov. 14, 2012

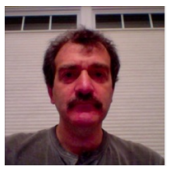
Acknowledgments

- Igor Griva - Mathematics, George Mason University

- Zi-Kui Liu - Materials Science, Pennsylvania State University



- Claudio Torres - Mathematics, George Mason University



- Dmitry Golovaty - Mathematics, University of Akron

- Shlomo Ta'asan - Mathematics, Carnegie Mellon University

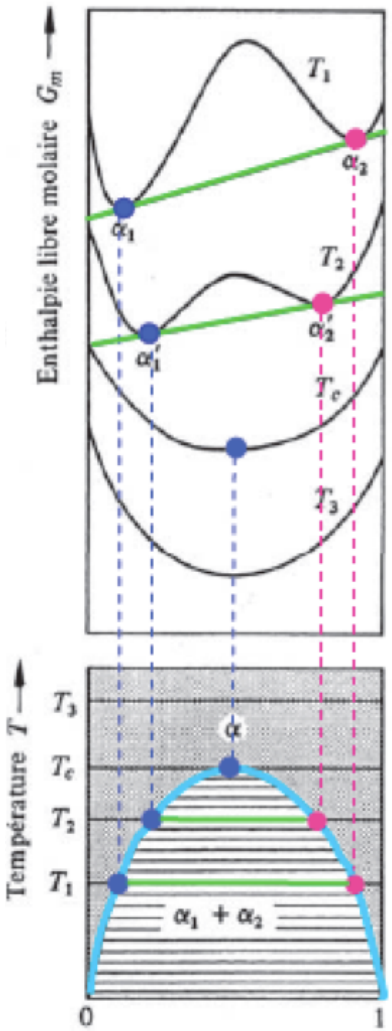
- David Kinderlehrer - Mathematics, Carnegie Mellon University

NSF CAREER DMS-1056821 and DMS-0915013

Outline of this talk

1. Problem 1: Phase diagram calculation
   - Outstanding challenges
   - Constrained optimization methods
   - AMPL model development
2. Problem 2: Microstructure evolution
   - Coarsening in microstructure: focus on critical events
   - Vertex models: numerical features and analysis
   - Mesoscale modeling: 1-dmensional model and its implications
3. Lessons learned and future directions

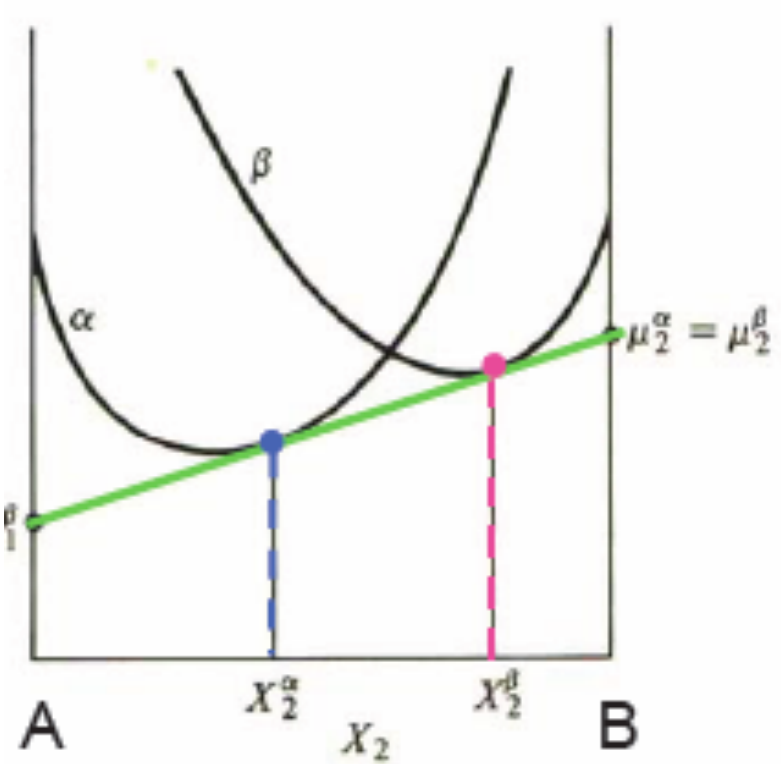
Phase diagram calculation



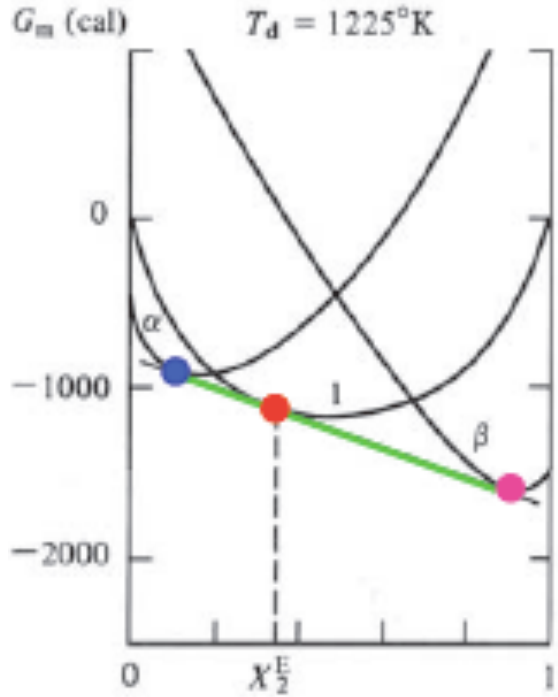
- Amounts to constructing common tangents
- Miscibility gaps need special treatment

Phase diagram calculation

2-phase equilibria

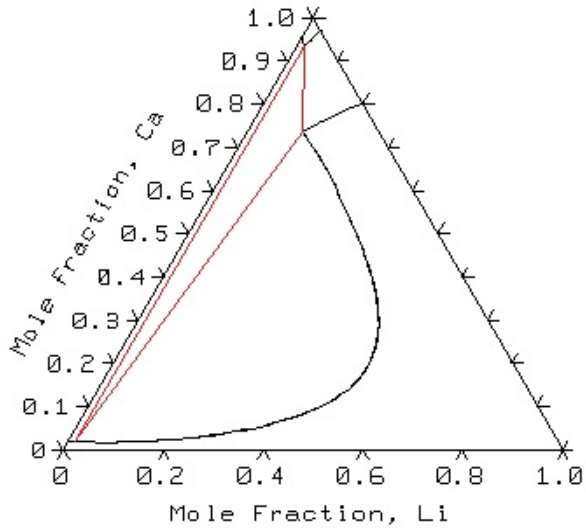


3-phase equilibria

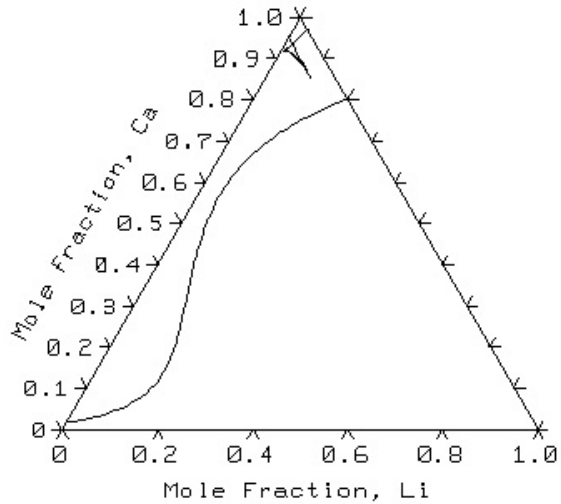


Can we trust what we get?

Correct diagram



"Naive" Thermocalc user would get this

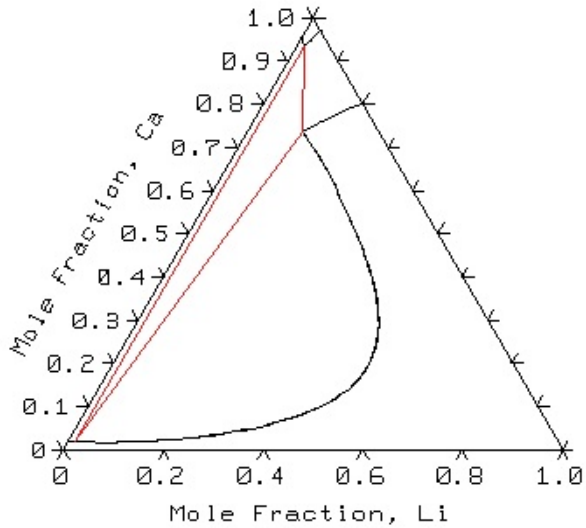


What is wrong?

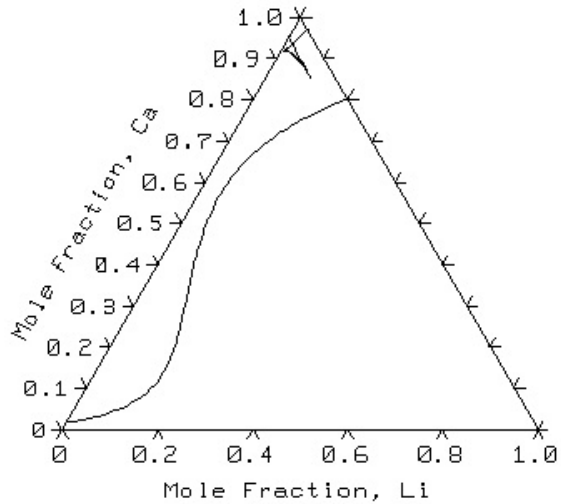
- Bad initial point
- Lack of a priori information about the Gibbs energy shape
- Special treatment is also required for multiple sublattice phases

Can we trust what we get?

Correct diagram



"Naive" Thermocalc user would get this



What is wrong?
- Bad initial point
- Lack of a priori information about the Gibbs energy shape
- Special treatment is also required for multiple sublattice phases

What can we do?

Automate the calculation to make it independent of:

- the number of subattices
- the initial guess
- the number of components

Our approach:
Develop one unified formulation for Gibbs energy minimization and use state-of-the-art methods to solve the constrained minimization problem.

Phase diagram calculation methods

Given individual phase energies $G^{(i)}$, phase diagram calculation amounts to solving the following optimization problem:

$$\min_{\mathbf{y},\mathbf{f}} \left\{ G = \sum_{i=1}^{N} f^{(i)} G^{(i)}(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) \right\}$$

$$\sum_{i=1}^{N} f^{(i)} x_j^{(i)} = f_j^0 \qquad j = 1, \ldots, K, j \neq Va$$

$$\frac{\sum_{k=1}^{M_i} a_k^i y_{k,j}^{(i)}}{\sum_{k=1}^{M_i} a_k^i (1 - y_{k,Va}^{(i)})} = x_j^{(i)} \qquad i = 1, \ldots, N, \quad j = 1, \ldots, K, j \neq Va$$

$$\sum_{j=1}^{K} y_{k,j}^{(i)} = 1 \qquad i = 1, \ldots, N, \quad k = 1, \ldots, M_i$$

For each phase $i = 1, \ldots, N$

$$\begin{cases} \mathbf{f}^{(i)} = \{f^{(i)}\}_{i=1,\ldots,N} & \text{- the phase fractions} \\ \mathbf{x}^{(i)} = \{x_j^{(i)}\}_{j=1,\ldots,K} & \text{- compositions} \\ \mathbf{y}^{(i)} = \{y_{k,j}^{(i)}\}_{k=1,\ldots,M_i;\ j=1,\ldots,K} & \text{- site fractions in sublattices} \end{cases}$$

Drawbacks of traditional approach

CALPHAD packages (Thermocalc, PANDAT, MTDATA, ChemSage etc) recast as an unconstrained optimization problem and solve the resulting linear system.

Shortcomings:

- Good initial guess required for convergence
- Till recently - no automatic miscibility gap detection
- Until now - inability to automate calculations in multiple sublattice case

New approach: key ideas

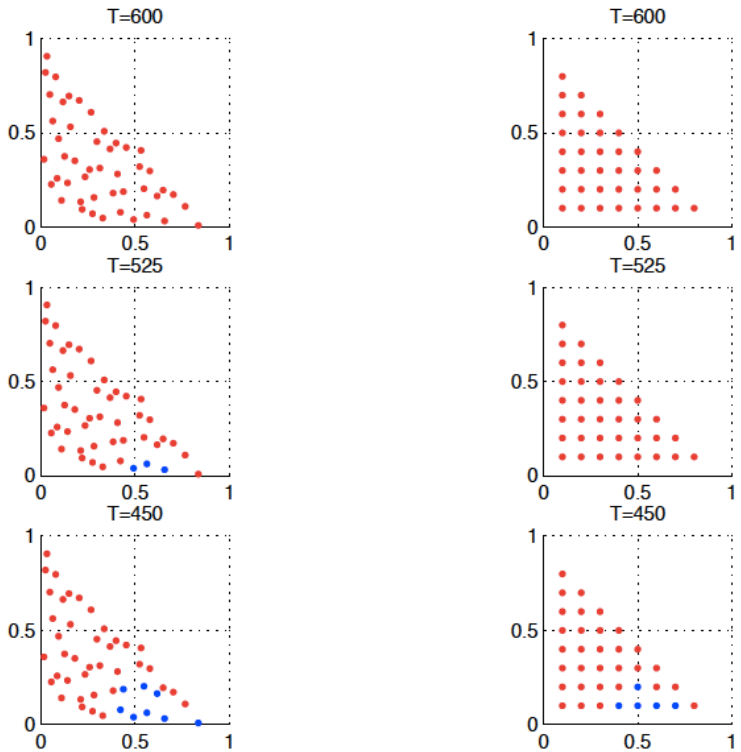
(Joint work with I. Griva, T. Stephens)

- Can use constrained optimization methods with less sensitivity to starting points

New approach: key ideas

(Joint work with I. Griva, T. Stephens)

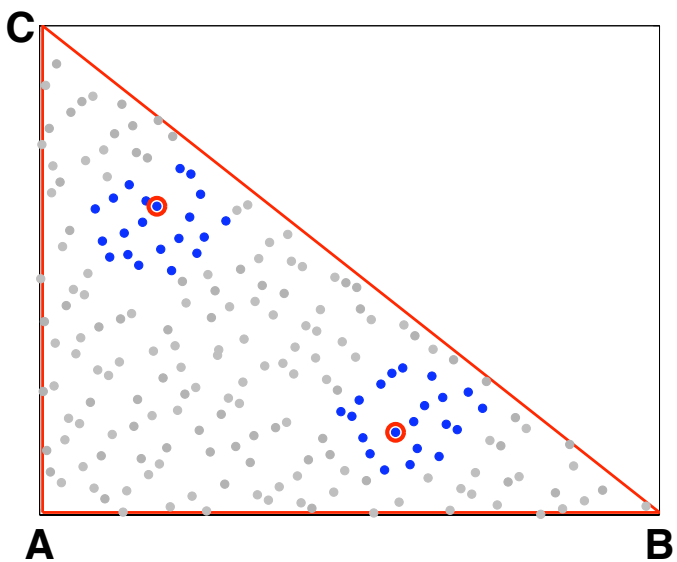
- Can use constrained optimization methods with less sensitivity to starting points
- Quasi-random sampling detects concavity better



New optimization approach

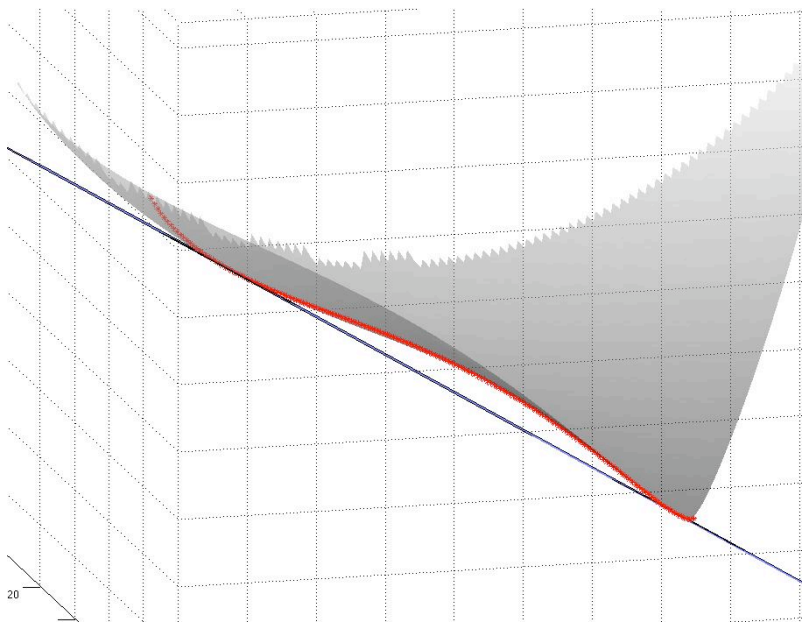
Developed for binary systems in [E.,Du, Liu, *Comp.Mat.Sci.*, 2005].

- Use efficient sampling approaches to determine regions of positive concavity for each of the phases (we use quasi-random sequences, e.g. Hammersley, Sobol sequences). Record any miscibility gaps.
- Eliminate unstable phases (those whose Gibbs energies are not minimal at the given concentration)
- Pick starting positions from the list of stable phases above (two per each miscibility gap detected)



New optimization approach

- Use efficient sampling approaches to determine regions of positive concavity for each of the phases (we use quasi-random sequences, e.g. Hammersley, Sobol sequences). Record any miscibility gaps.
- Eliminate unstable phases (those whose Gibbs energies are not minimal at the given concentration)
- Pick starting positions from the list of stable phases above (two per each miscibility gap detected)
- Solve the constrained optimization problem using SNOPT method



Advantages of the new method

Features of the constrained optimization solver:

- SNOPT - large-scale nonlinear constrained optimization solver based on a SQP strategy.
- Each QP subproblem minimizes a quadratic model of a Lagrangian function subject to a linearization of the constraints.
- An augmented Lagrangian merit function is reduced along each search direction to ensure convergence from practically any starting point.

Outcomes:

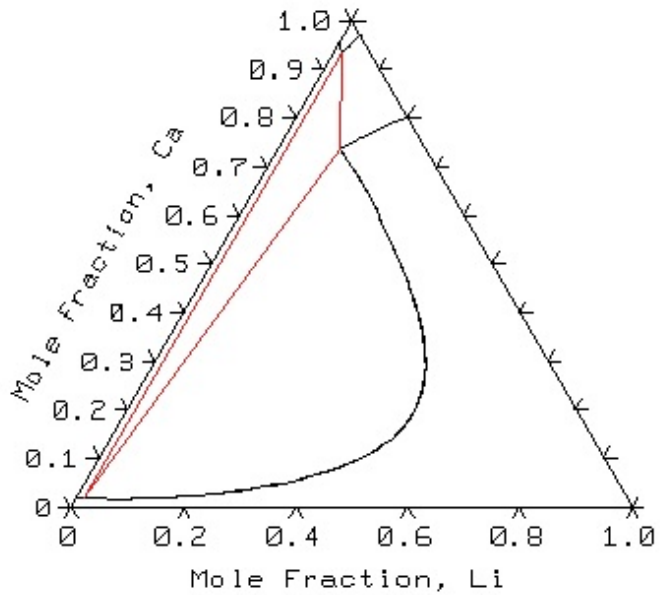
- Less sensitivity to initial guess $\Rightarrow$ a more robust numerical scheme
- Automatic handling of sublattices and miscibility gaps built into the code

CaLiNa system: automatic detection of miscibility gap

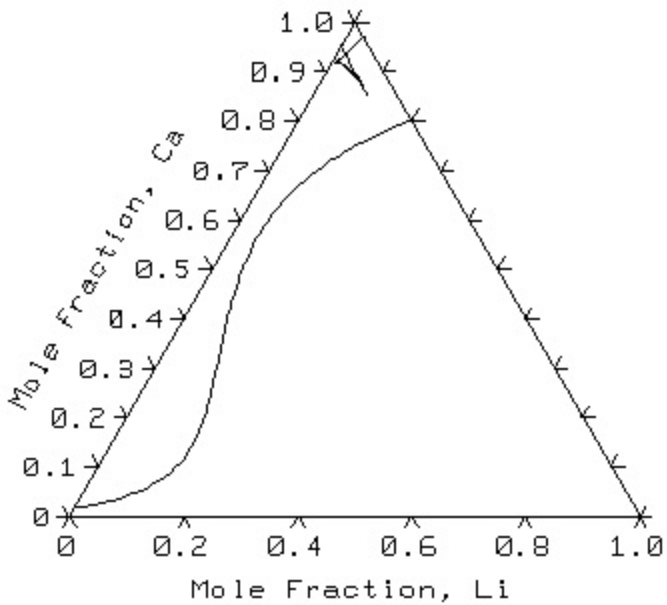
Numerical results show that:

- The new method arrives at the correct phase diagram in the presence of a miscibility gap

Correct diagram



Calphad result w/o user input

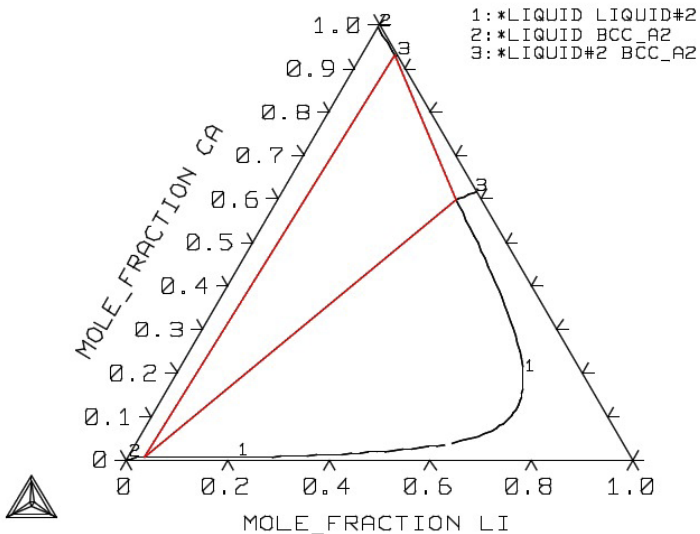


CaLiNa system: automatic detection of miscibility gap

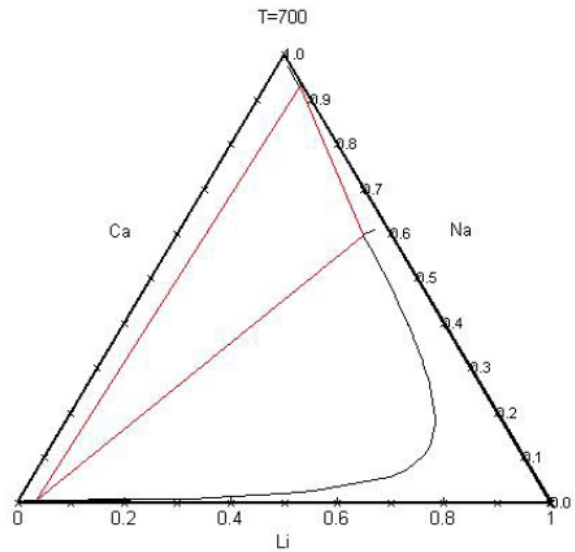
Numerical results show that:

- The new method arrives at the correct phase diagram in the presence of a miscibility gap

Correct diagram



Our result



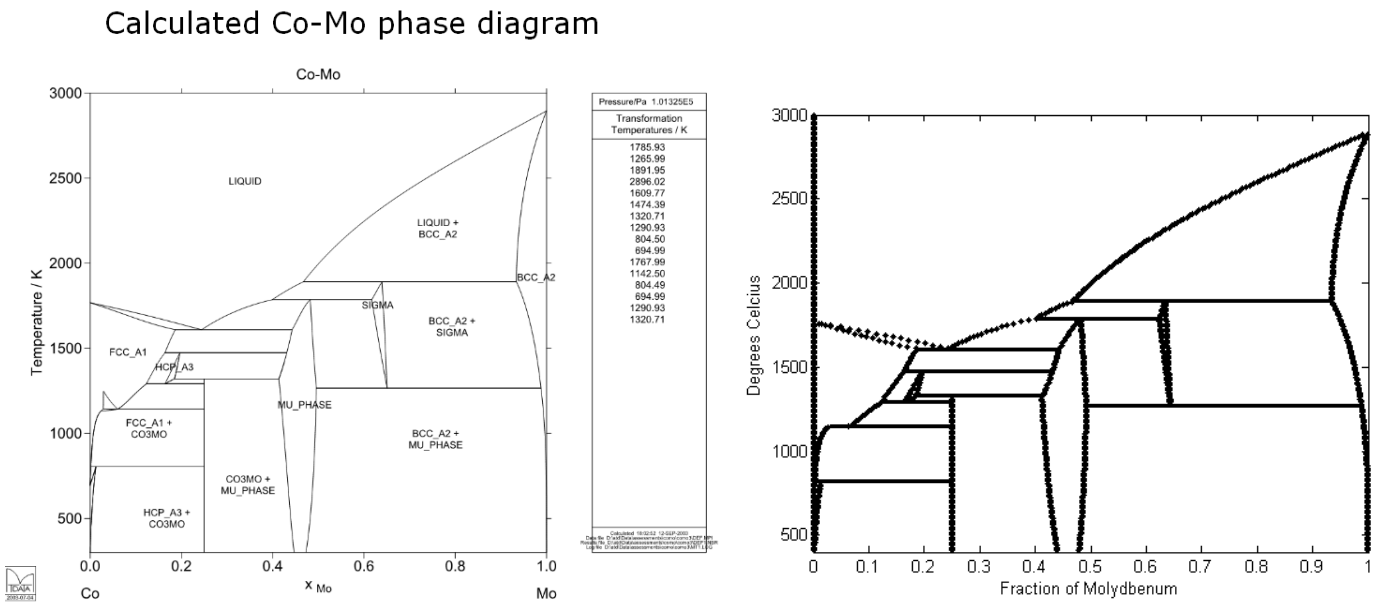
CoMo system: automatic handling of sublattices

Numerical results show that:

- The new method arrives at the correct phase diagram in the presence multiple sublattices

Correct diagram

Our result

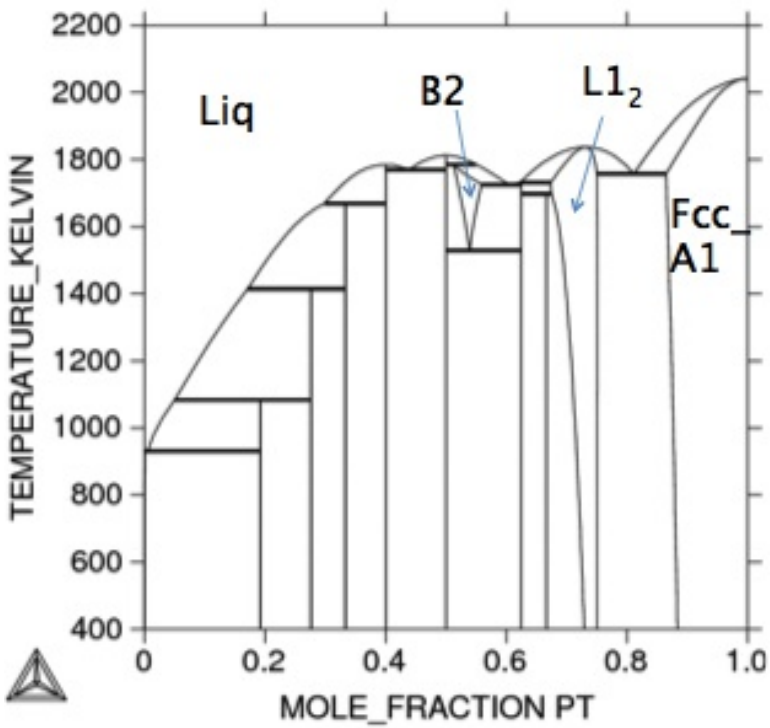


AlPt diagram: automation of sublattice case calculation

Numerical results show that:

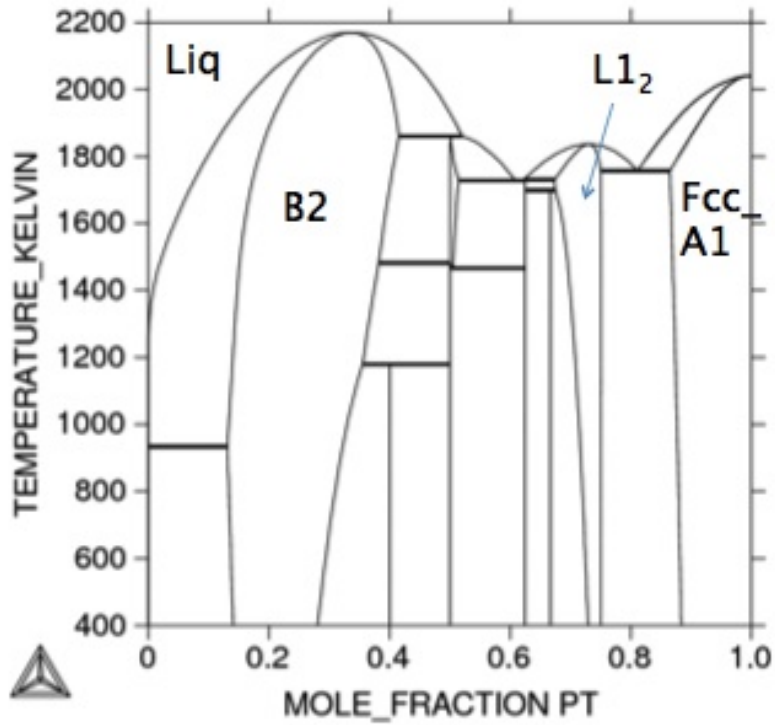
- The method handles sublattice phases better than standard option of Thermocalc, recovers missing phase information

Correct diagram



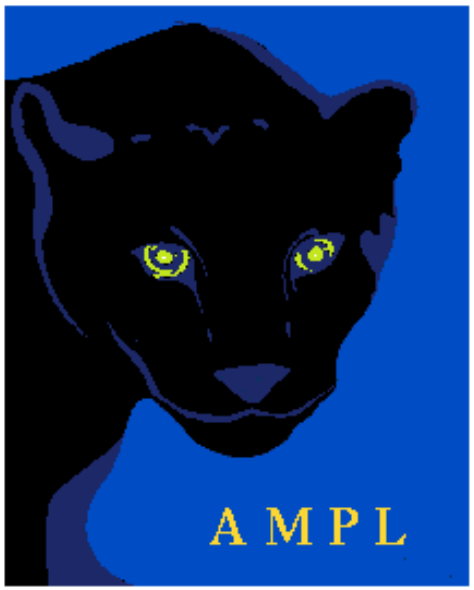
Special setting for sublattices

Calphad result w/o user input



Normal setting

AMPL environment



Key AMPL Features

Modeling language

Modeling environment

The AMPL book

Solvers that work with AMPL

Interfaces to popular and sophisticated solvers including CONOPT, CPLEX, LAMPS, LANCELOT, LOQO, LSGRG, MINOS, OSL, SNOPT, and XA.
Goal: to build a bridge between Thermocalc database and AMPL - regardless of the number of components, sublattices and phases.

Automatic Thermocalc to AMPL conversion

Thermocalc data contains information about all the phases and Gibbs energies in a specific format:

```
TYPE_DEFINITION ' GES A_P_D BCC_A2 MAGNETIC  -1.0    4.00000E-01 !
PHASE BCC_A2  %'  2 1   3 !
   CONSTITUENT BCC_A2  :AL,PT,VA : VA :  !

 PARAMETER G(BCC_A2,AL:VA;0)  2.98150E+02  +2106.85+132.280038*T
-24.3671976*T*LN(T)-.001884662*T**2-8.77664E-07*T**3+74092*T**(-1);
 7.00000E+02  Y
 -1193.24+218.235446*T-38.5844296*T*LN(T)+.018531982*T**2
-5.764227E-06*T**3+74092*T**(-1);  9.33470E+02  Y
 -1195.378+183.871153*T-31.748192*T*LN(T)-1.230524E+28*T**(-9);
 2.90000E+03  N REF1 !
 PARAMETER G(BCC_A2,PT:VA;0)  2.98150E+02  +7404.369+121.988275*T
-24.5526*T*LN(T)-.00248297*T**2-2.0138E-08*T**3+7974*T**(-1);  1.30000E+03
 Y
 +5746.826+159.129615*T-30.2527*T*LN(T)+.002321665*T**2-6.56946E-07*T**3
-272106*T**(-1);  2.04150E+03  Y
 -207048.216+1016.95892*T-136.192996*T*LN(T)+.020454938*T**2
-7.59259E-07*T**3+71539020*T**(-1);  4.00000E+03  N REF1 !
```

Automatic Thermocalc to AMPL conversion

Thermocalc database

```
TYPE_DEFINITION ' GES A_P_D BCC_A2 MAGNETIC  -1.0    4.00000E-01 !
PHASE BCC_A2  %'  2 1   3 !
   CONSTITUENT BCC_A2  :AL,PT,VA : VA :  !

  PARAMETER G(BCC_A2,AL:VA;0)  2.98150E+02  +2106.85+132.280038*T
 -24.3671976*T*LN(T)-.001884662*T**2-8.77664E-07*T**3+74092*T**(-1);
 7.00000E+02  Y
  -1193.24+218.235446*T-38.5844296*T*LN(T)+.018531982*T**2
 -5.764227E-06*T**3+74092*T**(-1);  9.33470E+02  Y
  -1195.378+183.871153*T-31.748192*T*LN(T)-1.230524E+28*T**(-9);
 2.90000E+03  N REF1 !
  PARAMETER G(BCC_A2,PT:VA;0)  2.98150E+02  +7404.369+121.988275*T
 -24.5526*T*LN(T)-.00248297*T**2-2.0138E-08*T**3+7974*T**(-1);  1.30000E+03
  Y
  +5746.826+159.129615*T-30.2527*T*LN(T)+.002321665*T**2-6.56946E-07*T**3
 -272106*T**(-1);  2.04150E+03  Y
  -207048.216+1016.95892*T-136.192996*T*LN(T)+.020454938*T**2
 -7.59259E-07*T**3+71539020*T**(-1);  4.00000E+03  N REF1 !
```

⇒

AMPL description

Automatic Thermocalc to AMPL conversion

All data in AMPL is stored in SETS, as follows:

[bcc_a2, *, 1 ,*]: #1 is Platinum

	1	2	3 :=	
T_UB	1300	2041.5	4000	
a0	7404.369	5746.826	-207048.21	#CONSTANT
a1	121.988275	159.129615	1016.95892	#T
a2	-24.5526	-30.2527	-136.192996	#TlogT
a3	-0.00248297	0.002321665	0.020454938	#T**2
a4	7974	-272106	71539020	#T**-1
a5	-2.0138E-08	-6.56946E-07	-7.59259E-07	#T**3
a6	0	0	0	#T**7
a7	0	0	0	#T**-9
b0	0	0	0	#GHSERAL
b1	0	0	0	#GHSERPT
b2	0	0	0	#GBCCAL
b3	0	0	0	#GBCCPT
b4	0	0	0	#B2ALVA
b5	0	0	0	#LB2ALVA
b6	0	0	0	#B2PTVA
b7	0	0	0	#LB2PTVA
b8	0	0	0	#UN_ASS

Summary of phase diagram calculation findings

- Converter is able to transfer any phase data from Thermocalc into powerful optimization environment AMPL
- An automatic phase diagram can be generated via the AMPL script able to handle arbitrary muticomponent systems
- Description provides a way to handle: multiple sublattices, miscibility gaps, order-disorder transitions
- Multiple optimization routines can be applied to the same system to compare the optimal solutions found and improve and complete current database descriptions

Conclusion

The new approach has a potential for giving a more accurate information about location of phase transition zones.

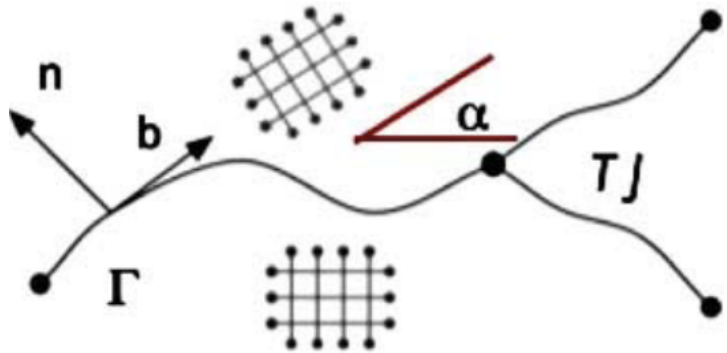
Databases have to be re-evaluated for possibility of missing phases, like in the case above.

PROBLEM 2: Microstructure evolution

Polycrystals:
network of crystals (grains) separated by interfaces (grain boundaries)
Grain growth:
coarsening of the network under external heating

(Loading movie)

Mathematical formulation: notations



- GB is a curve $\Gamma : x = \chi(s)$, $0 \leq x \leq L$ with
- $b = \dfrac{d\chi}{ds}$ (tangent) and $n = Rb$(normal), where R =rotation through $\pi/2$.
- α is the lattice misorientation parameter
- $\gamma = \gamma(n, \alpha) > 0$ is the surface tension (surface energy density) of Γ
- the energy of Γ is given by $E = \int_\Gamma \gamma(n, \alpha)|b|\, ds$, the amount of work required to generate an infinitesimal amount of new surface

Straight-line approximation of boundaries

$$E = \Sigma_j \int_{\Gamma^{(j)}} \gamma(n^{(j)}, \alpha^{(j)}) |b^{(j)}| \, ds$$

Simplification: $\gamma = \gamma(\alpha)$ means all derivatives of γ drop out.
Consequences:

$$E = \int_{\Gamma} \gamma(\alpha) |b| \, ds = \sum_j \gamma_j(\alpha_j) |x_1^j - x_2^j|$$

$$\frac{dE}{dt} = \sum_m \dot{x}_m \Big(\sum_l -\gamma_{ml} b_{ml}(t) \Big)$$

So the TJs move according to the sum of line tensions:

$$\dot{x}_m(t) = \sum_l \gamma_{ml} b_{ml}(t)$$

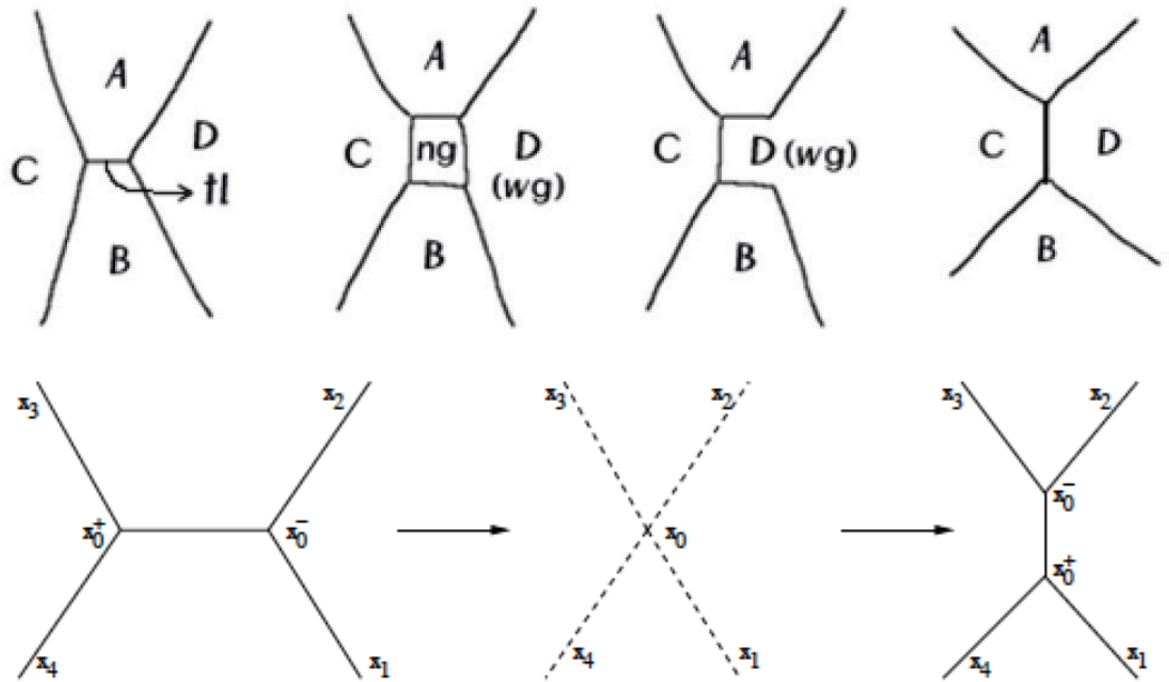
Vertex model -

a special case of the curvature-driven growth evolution, where the GB mobilities are assumed to be significantly higher than TJ mobilities. Similar to a TJ drag.

Reconfiguration events

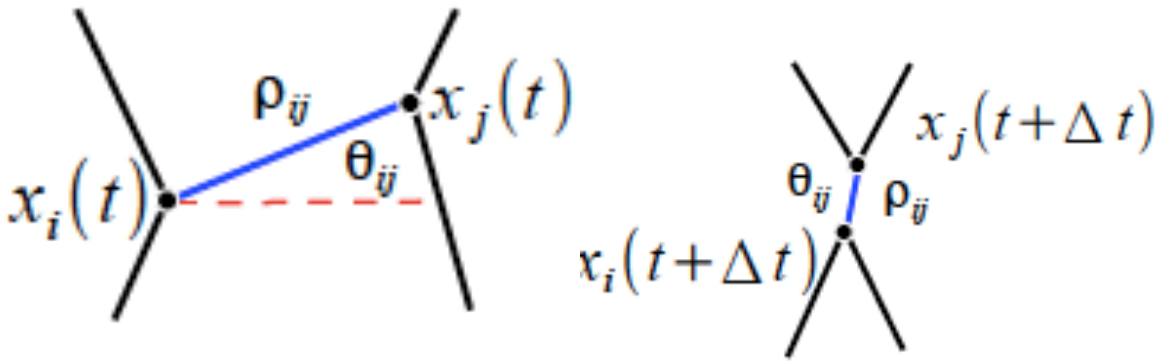
In addition to continuous evolution, grain boundary network experiences the following topological changes during grain growth:

- Loss of a facet (facet flipping)
- Loss of a small grain



Simulations differ in treatment of these events.

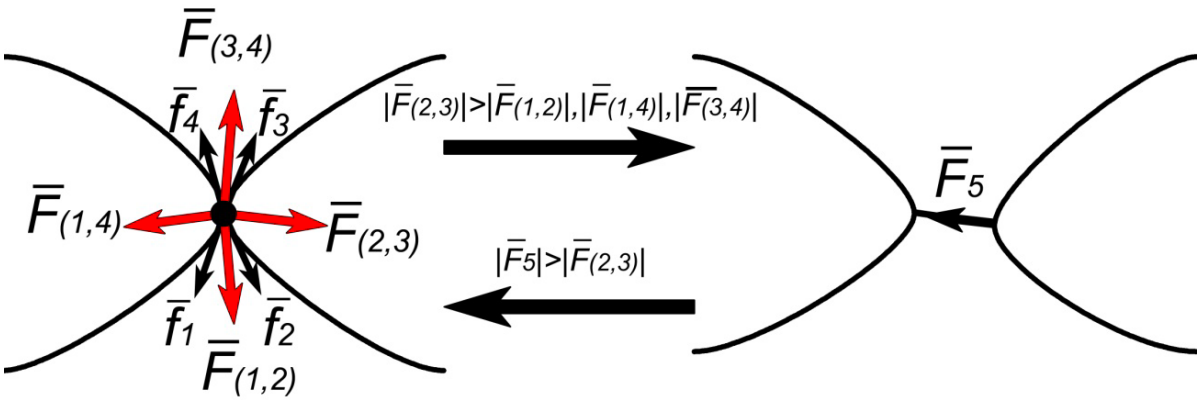
TJ motion during the reconfiguration event

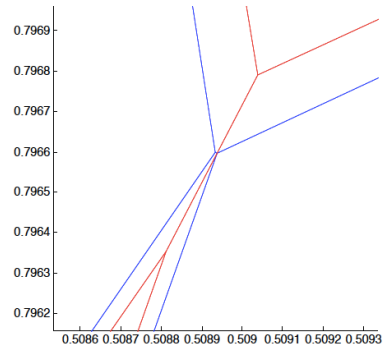


From the evolution of the TJ, we get:

$$\dot{\rho}_{ij} = -2 + C\cos\theta_{ij} + S\sin\theta_{ij},$$
$$\dot{\theta}_{ij} = \frac{1}{\rho_{ij}}(S\cos\theta_{ij} + C\sin\theta_{ij})$$

Different flipping rules





1. Maximum energy dissipation [Ta'asan, Yu, Livshits, Kinderlehrer, Lee 2006]
2. Maximum resulting force: in the direction of $F(2,3)$ [Barrales Mora 2010]
3. Fastest new edge growth rate: in the direction of $F(2,3) - F(1,4)$ [E., Golovaty, Torres]
4. Other versions of vertex models: Soares (85), Kawasaki (89), Otto/Niethammer (04)

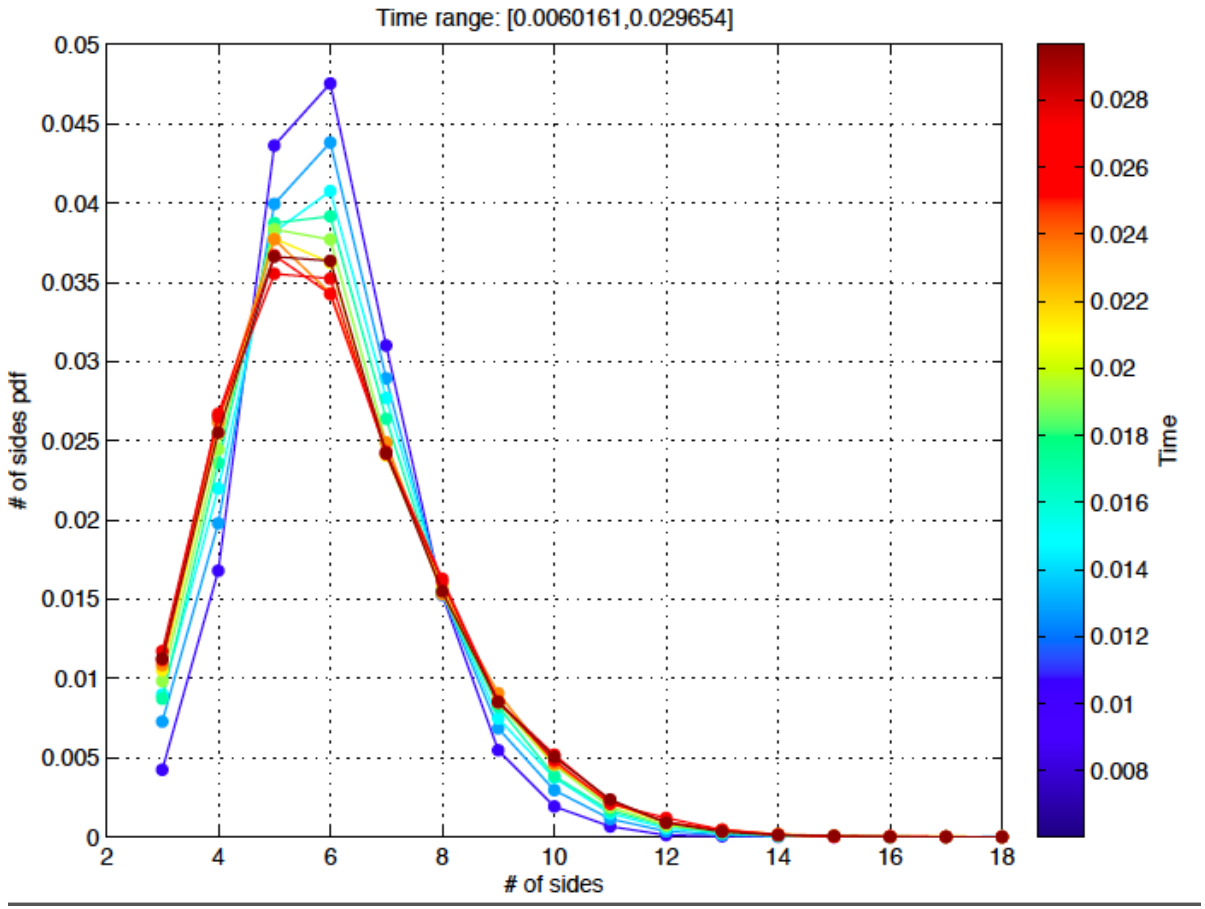
Our analysis shows that the most natural condition is #3.

Our vertex simulation

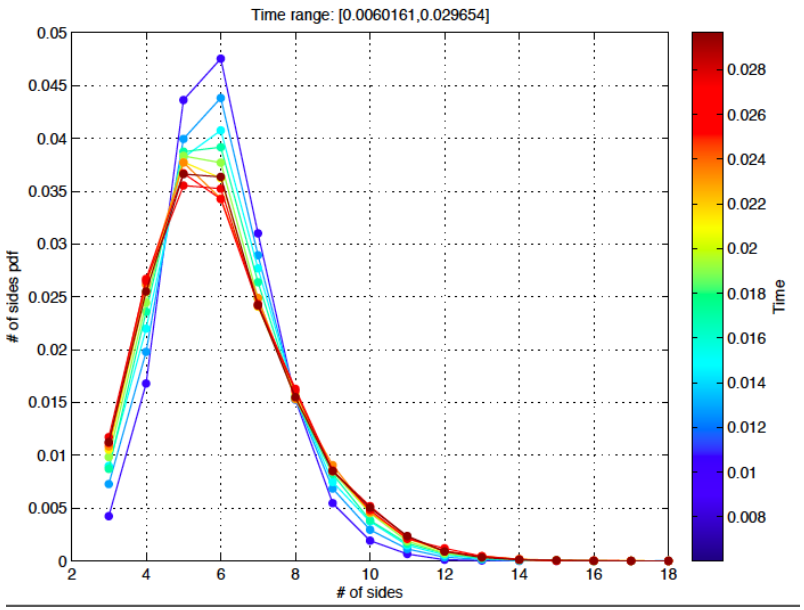
1024 grains, $\gamma(\alpha) = 1 + 0.02(1 - cos(2\pi\alpha)^3)$. Periodic boundary conditions.

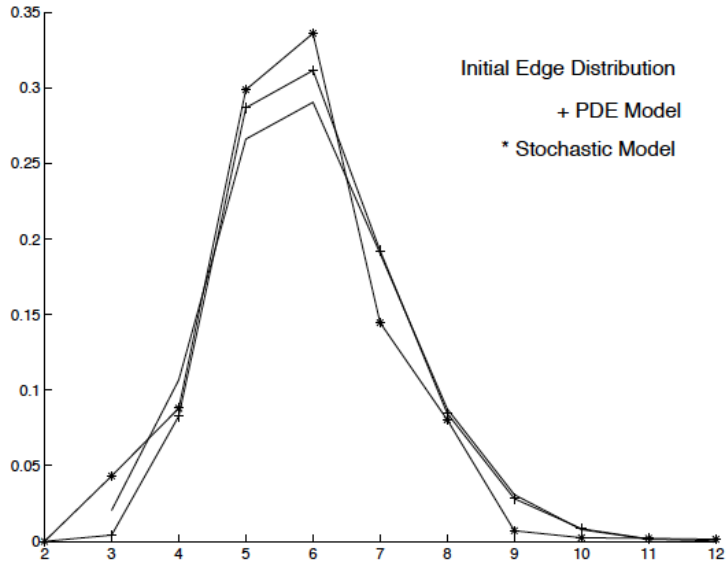
(Loading movie)

Vertex code statistics: n-sided grain pdf



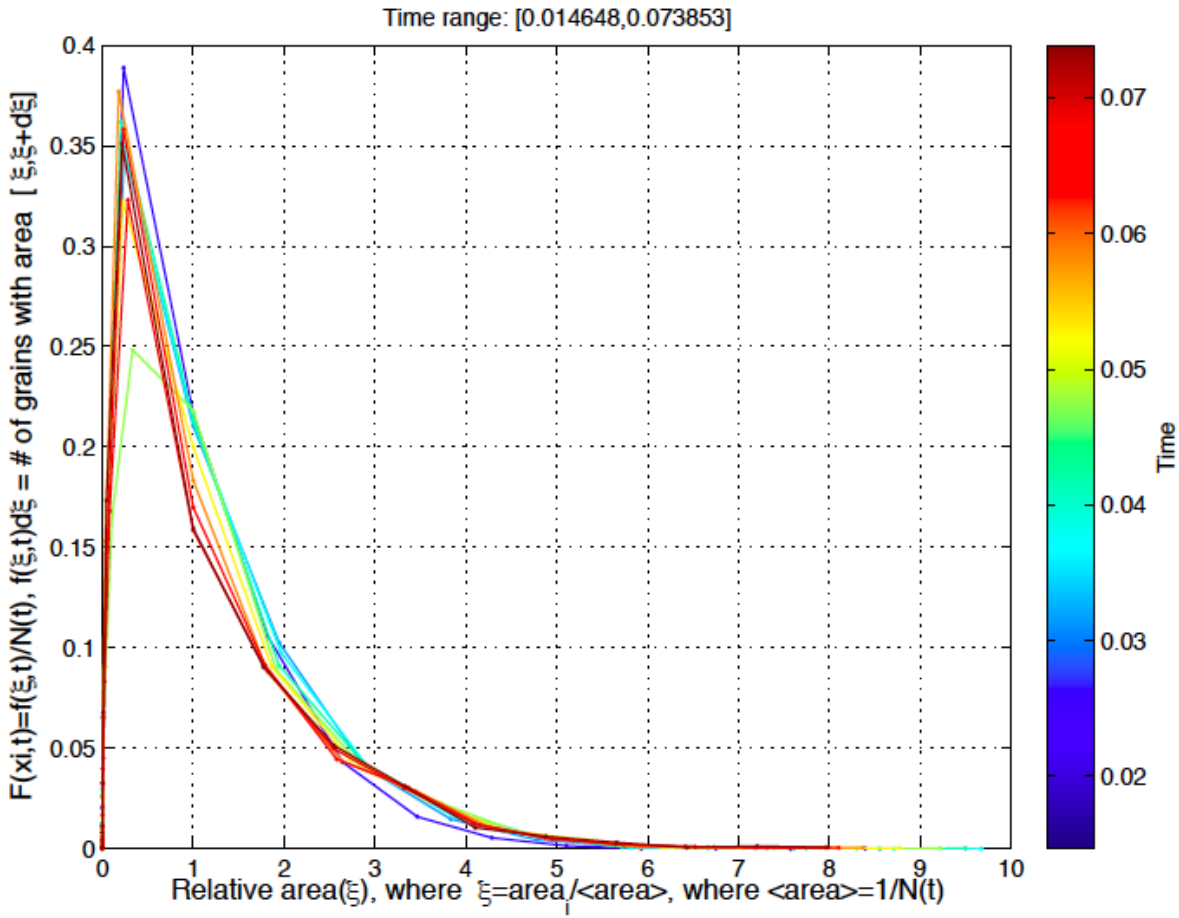
Vertex code statistics: n-sided grain pdf



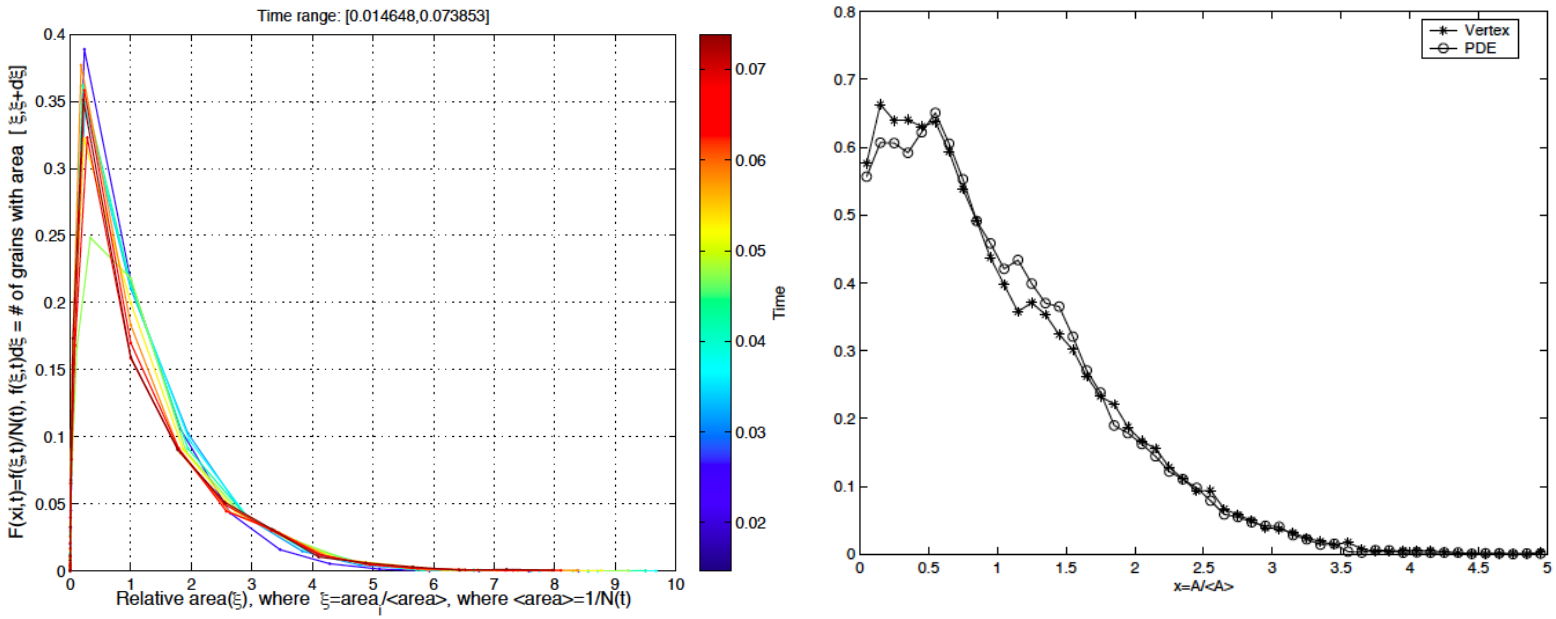


N-sided statistics matches that produced by curvature-driven PDE and Monte Carlo models.

Vertex code statistics: relative area pdf

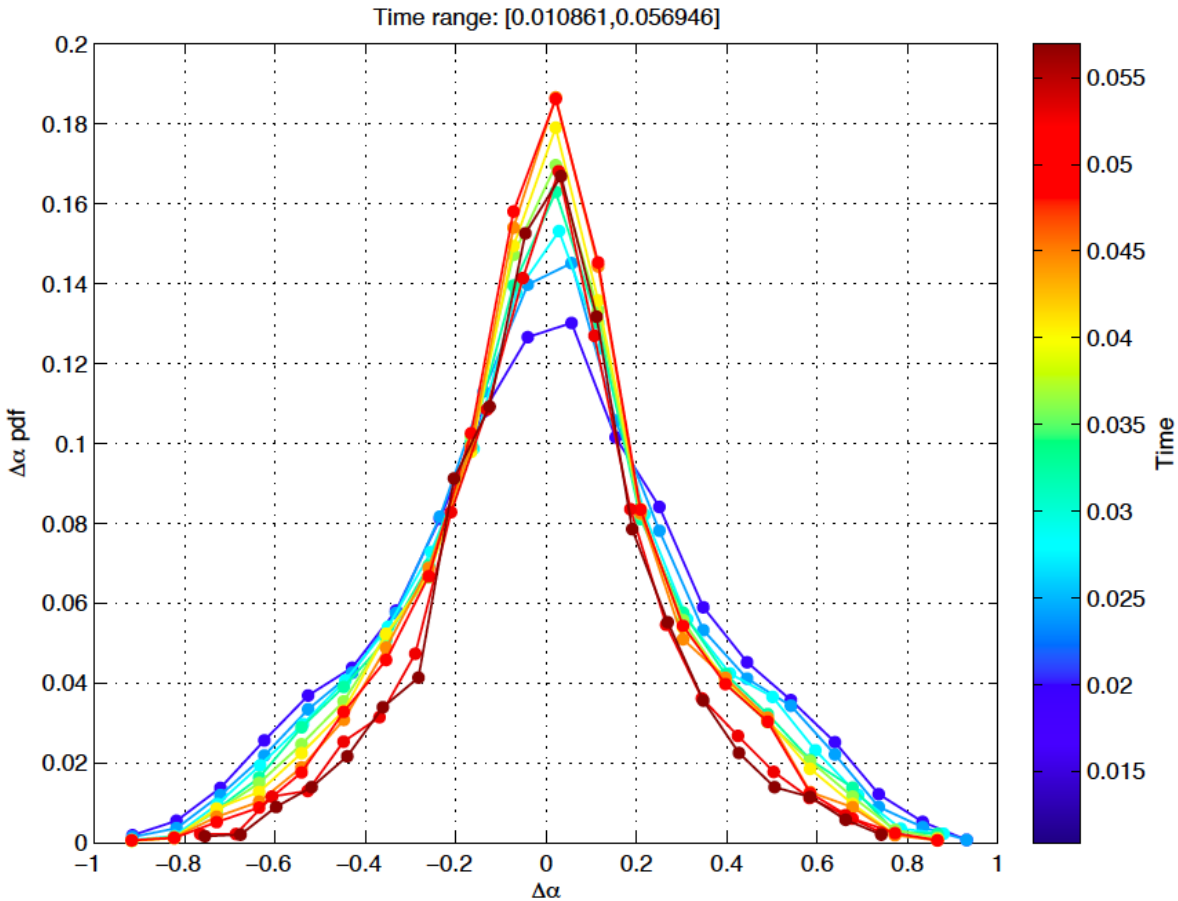


Vertex code statistics: relative area pdf

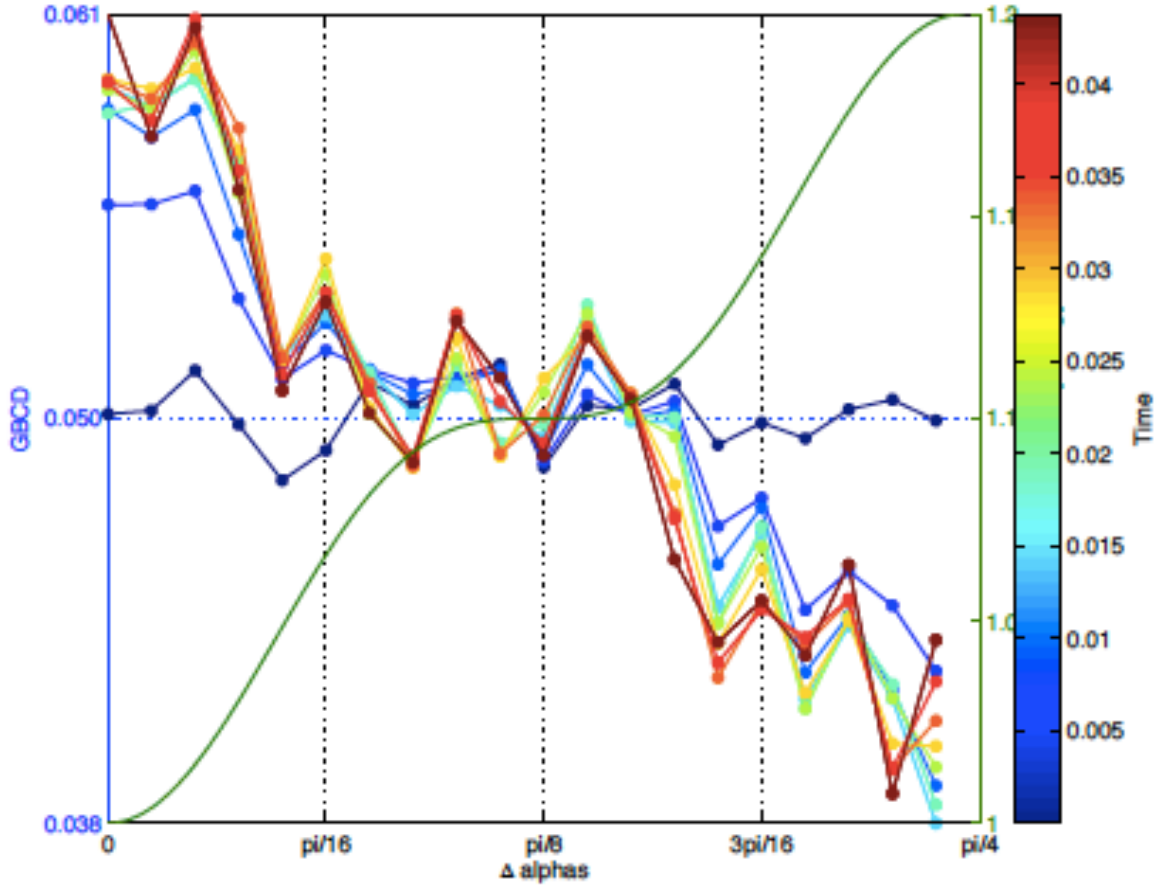


Area statistics is similar to that produced by GB-driven PDE and Monte Carlo models, while the peak is somewhat higher.

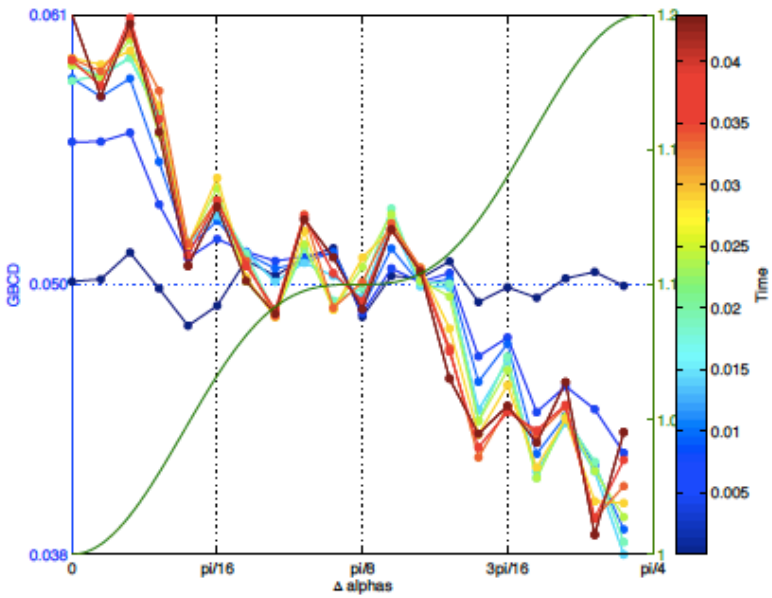
Vertex code statistics: misorientations pdf

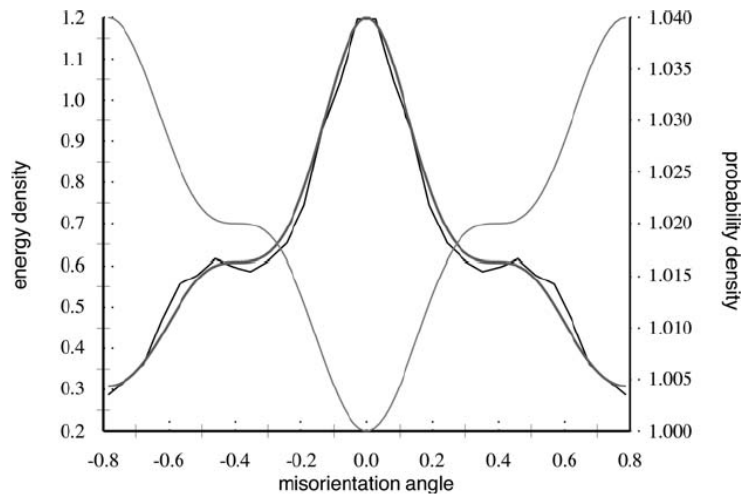


Vertex code statistics: GBCD



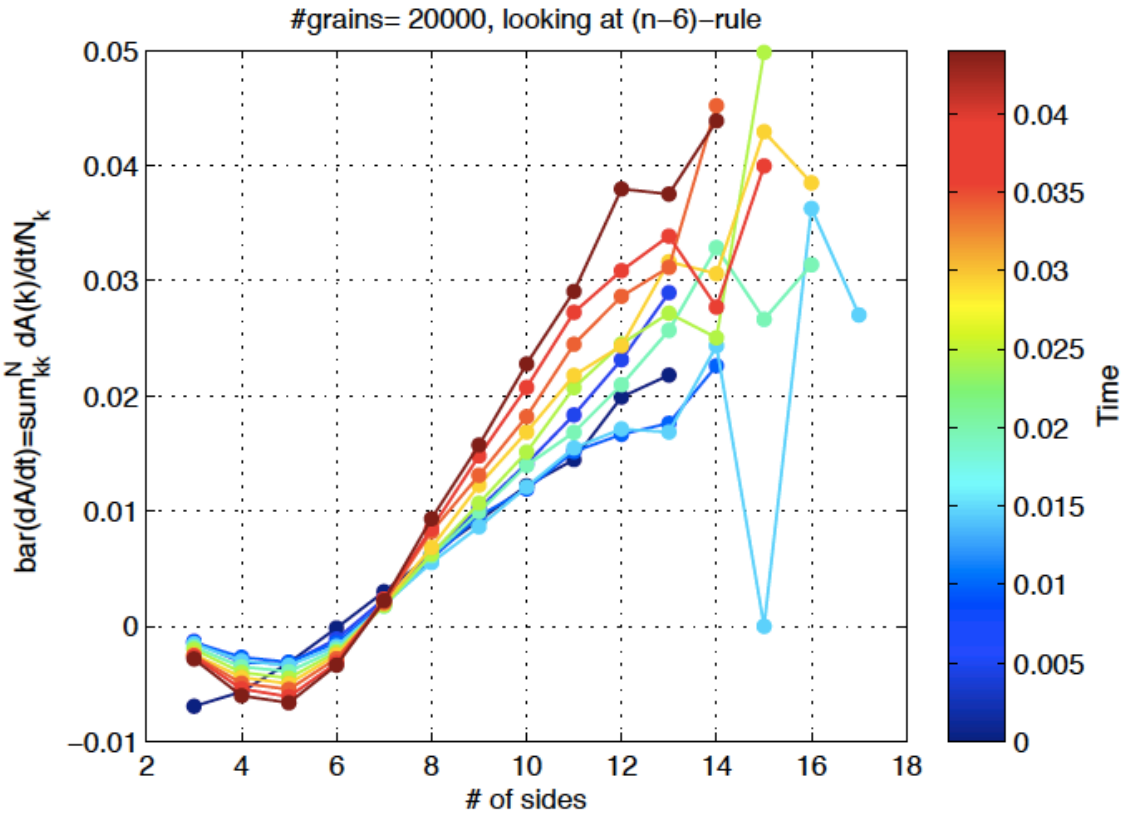
Vertex code statistics: GBCD





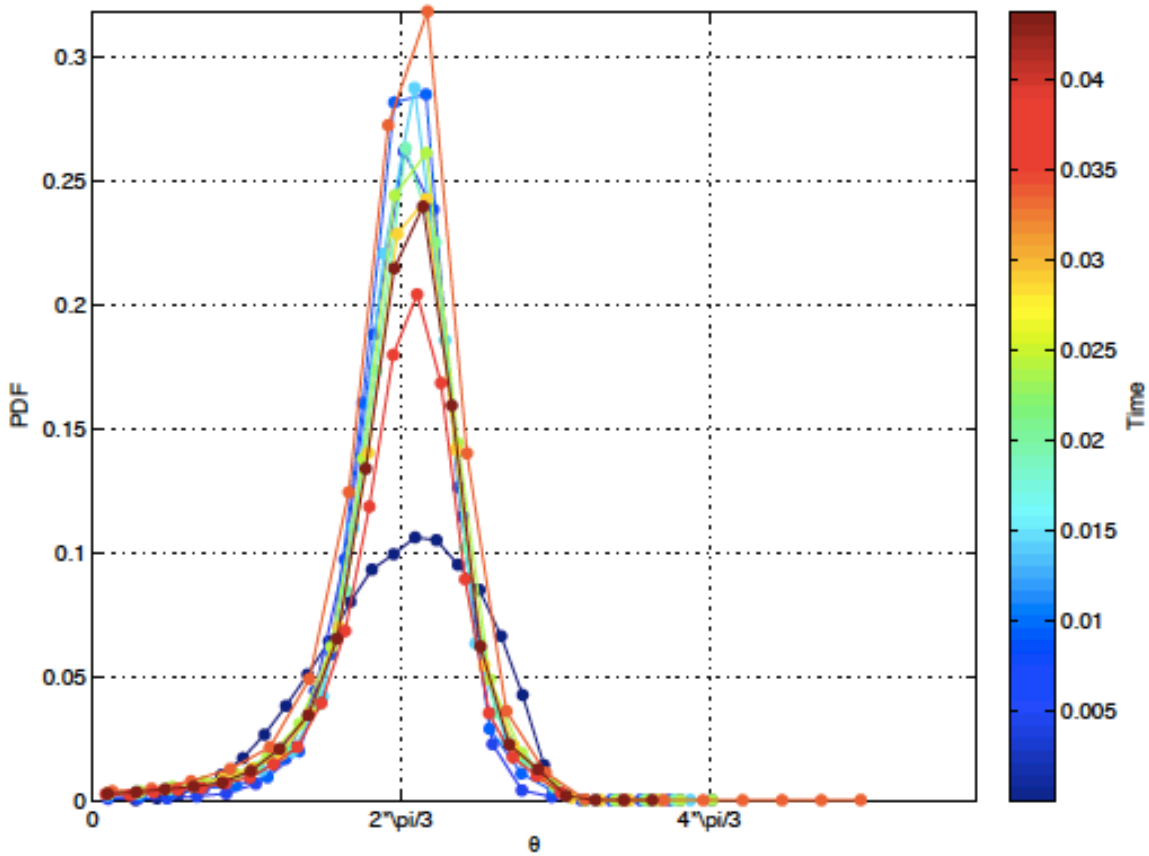
If averaged over many trials, GBCD statistics is similar to GB-driven models.

Vertex code statistics: n-6 rule



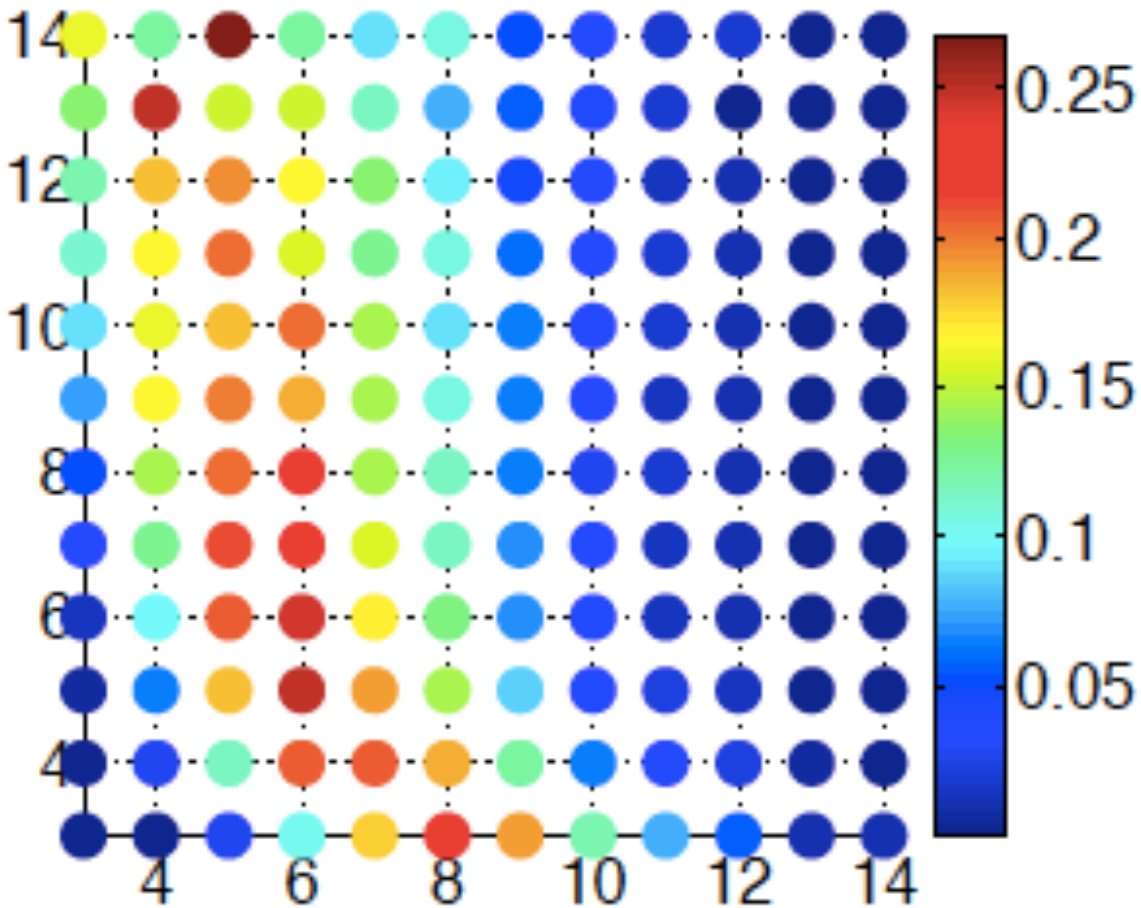
Mullins - von Neumann "n-6" rule does not hold for grains with less than 6 sides. This is affected by the treatment of the small grain disappearance events.

Vertex code statistics: angle distribution



Distribution of angles at triple junctions is wider than in GB-riven simulations.

Vertex code statistics: neighbor correlations



Probability of an i-sided grain neighboring a j-sided grain. Same as what is expected.

Manifestation of unstable behavior

(Loading movie)

Accuracy and stability analysis

Critical discretization parameters:

- Δx_0 - the minimal edge length
- Δt - time step for the numerical scheme

Theoretical estimates:

- Collision time:

$$t_\lambda = \frac{\rho_{ij}(0)(\dot{\rho}_{ij}(0) + 4)}{4 - ||p||^2}$$

  where $p = F(2,3) - F(1,4)$.

- Eigenvalues of the local Jacobian scale as $1/\rho_{ij}$. Hence computation is unstable for small ρ_{ij}.

These estimates allow to define Δt and Δx_0 in a correct way to avoid stability issues.

Preliminary results of the vertex model analysis

Observations so far:

- Vertex codes do recover the shape of all the main statistics produced by curvature codes
- Herring condition is satisfied only in the limit
- Models vary with respect to their 'n-6 rule' performance, which depends on the treatment of small grains
- Flipping rules vary dramatically among codes. This doesn't matter for pdfs, but matters for dynamics
- Resolution has a quantifiable effect on the accuracy and stability

What's ahead:

- Other major differences between GB vs. TJ driven evolution?
- How does discretization accuracy of the GBs affect the statistics?
- Do vertex codes preserve correlation statistics among misorientations/velocities?
- Extension of vertex models to include dependence on the normal
- Extensions to higher dimensions

More details in C. Torres poster!

Modeling strategies inspired by TJ motion and reconfiguration events

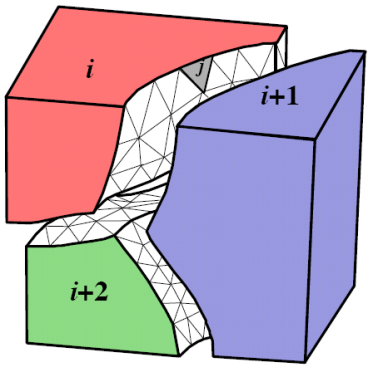
There have been several mesoscale models that attempt to describe the behavior of the distributions, based on the 1-dimensional model given below:

- Fractional random walk model
- Boltzmann model
- Entropy-based model (details in the talk by Y. Epshteyn today)

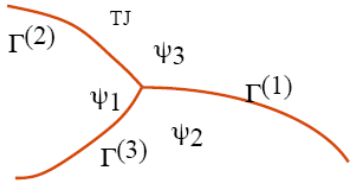
Provide different views on the same problem.

Simplified model

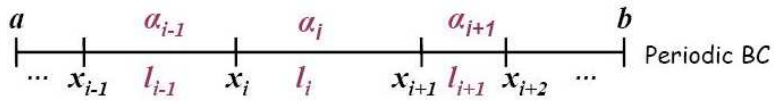
3D



2D



1D



α_i - misorientations
$\gamma(\alpha)$ - interfacial energies

x_j	–	"triple junctions"
(x_j, x_{j+1})	–	"grain boundaries"
$l_j = x_{j+1} - x_j$	–	"GB areas"
$\{\alpha_i\}_{i=1,\dots,n}$	–	"misorientations"

Given initial orientations α_i and total system energy in the form

$$E_n(t) = \sum \gamma(\alpha_i)(x_{i+1}(t) - x_i(t))$$

define equations of motion through gradient flow dynamics

$$\dot{x}_i = \gamma(\alpha_i) - \gamma(\alpha_{i-1}), \;\; i = 0, \dots, n.$$
$$v_i = \dot{l}_i = \dot{x}_{i+1} - \dot{x}_i = \gamma(\alpha_{i+1}) - 2\gamma(\alpha_i) + \gamma(\alpha_{i-1})$$

where $\gamma(\alpha)$ plays the role of an interfacial energy.

Influence of interfacial energies

Conforming to experimental observations, distribution of orientation parameters α is inversely correlated with interfacial energies $\gamma(\alpha)$.
Moreover - it matches 2-dimensional simulation results, and has a Boltzmann distribution shape. Cannot be a coincidence!

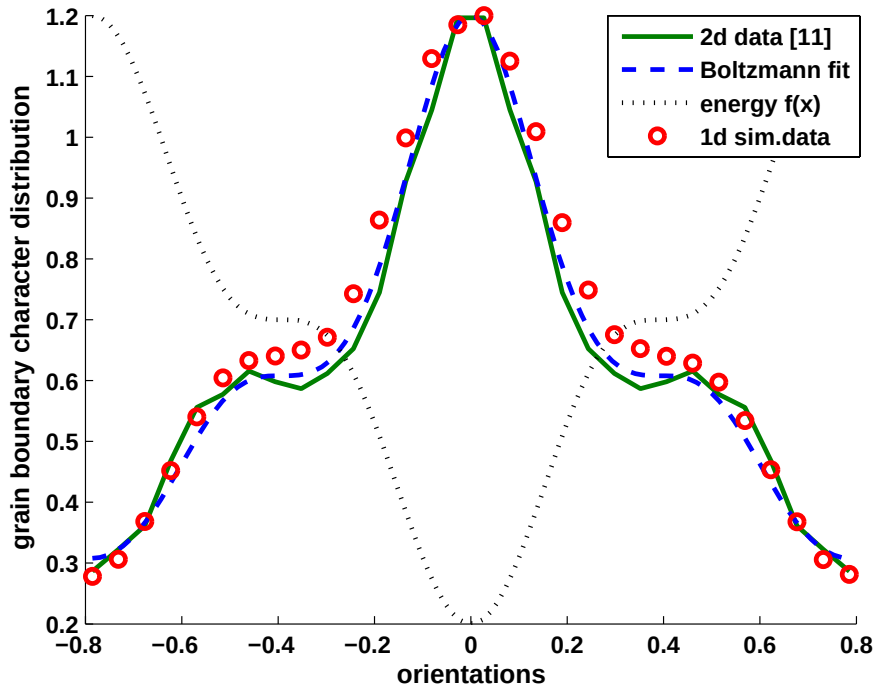
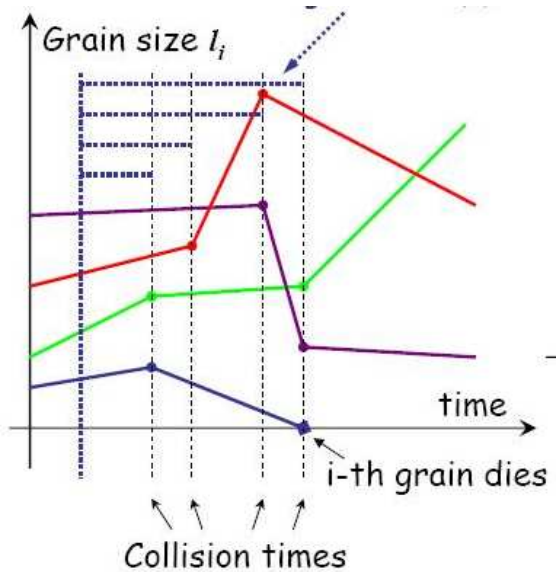


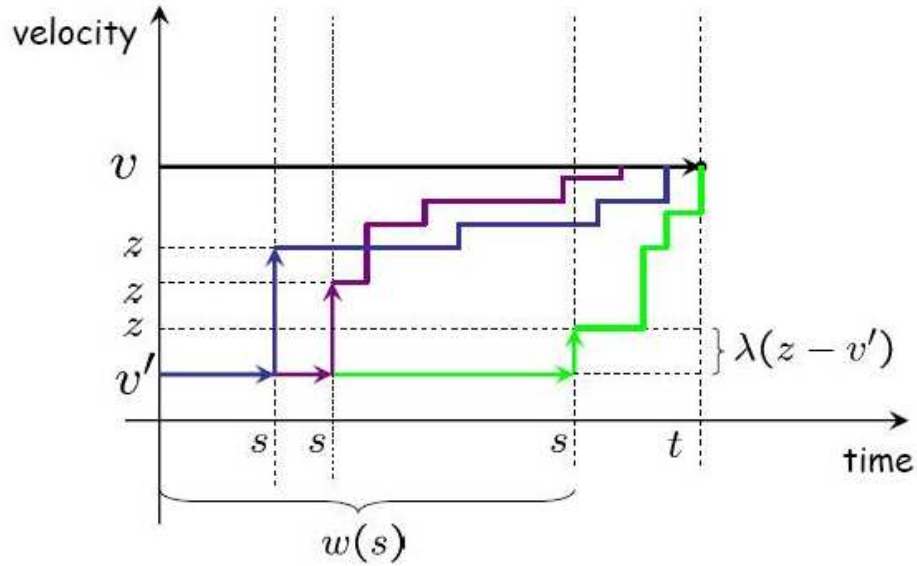
Figure: GBCD comparison.

Continuous time random walk of the lengths and velocities

Grain areas

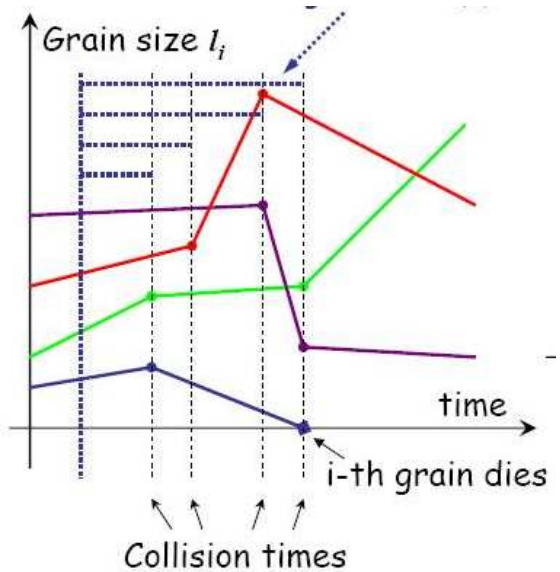


Grain velocities

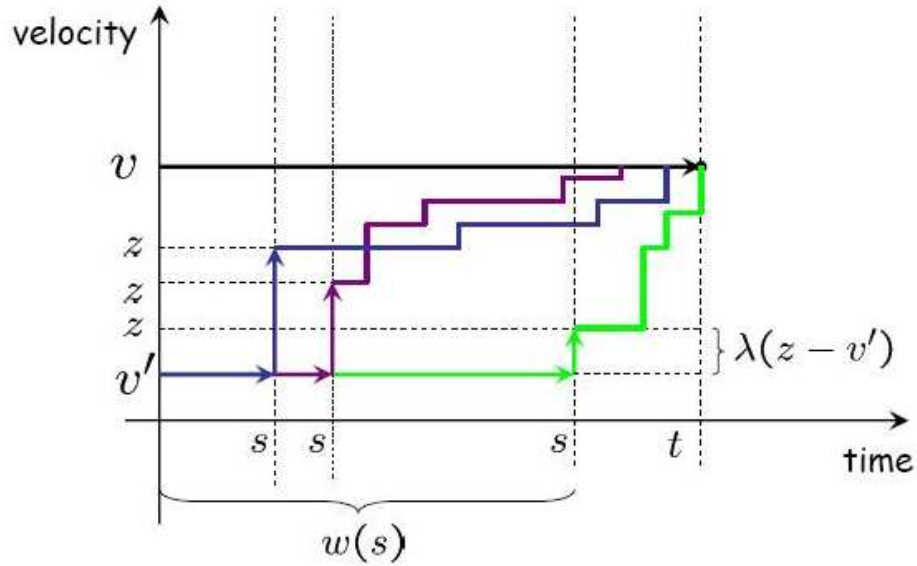


Continuous time random walk of the lengths and velocities

Grain areas

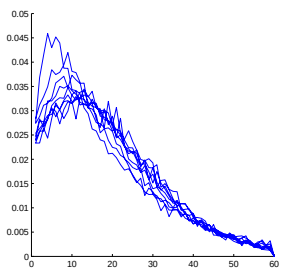


Grain velocities

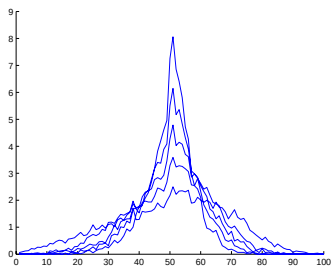


Simulation results: steady-state statistics of l_i and v_i (exists for all types of f)

relative areas



relative velocities



Velocity jump process of triple junctions

Rules of motion:

- RUN: TJ moves in one direction until it encounters another TJ
- REORIENTATION: The direction of motion changes after the collision

Resembles the motion of bacteria, but with no tumble:

Coarsening dynamics

What we know:

1. Coarsening process contains discontinuities (jumps) created by topological reconfigurations
2. The rate with which these events occur (arrival rate λ) goes down over time
3. Knowing the arrival rate, we can write the evolution equation for the mesoscale statistics

Idea:

- Use non-homogeneous velocity jump process to describe evolution of the statistics
- Make the arrival rate dependent on the state of the system

Mesoscopic evolution theory via Levy process theory

Stochastic process $X(t)$ in (l, v) space is a Levy process with triplet (b, A, ν).

$b = (v, 0)^T$ is the drift,
$A = 0$ is the diffusive part
$\nu = \mu(s)$ is the distribution of the jumps of v variable
λ is the intensity parameter

From Levy-Khintchine Theorem, it is easy to derive:

$$\frac{\partial \rho(l, v, t)}{\partial t} = v \frac{\partial \rho}{\partial l} + \lambda \int [\rho(l, v + s, t) - \rho(l, v, t)] \mu(s)\, ds$$

This model belongs to the class of nonlocal diffusion equations and has been shown to satisfy maximum principle, with no blow-up of solutions possible [Du et al, 2011].

State-dependent dynamics

Coupled system including internal state variable y:

$$\frac{\partial \rho(l, v, y, t)}{\partial t} = v\frac{\partial \rho}{\partial l} + \lambda \int [\rho(l, v+s, y, t) - \rho(l, v, y, t)]\mu(y, s)\, ds$$
$$= v\frac{\partial \rho}{\partial l} + \lambda \int \rho(l, v+s, y, t)\mu(y, s)\, ds - \rho(l, v, y, t)$$

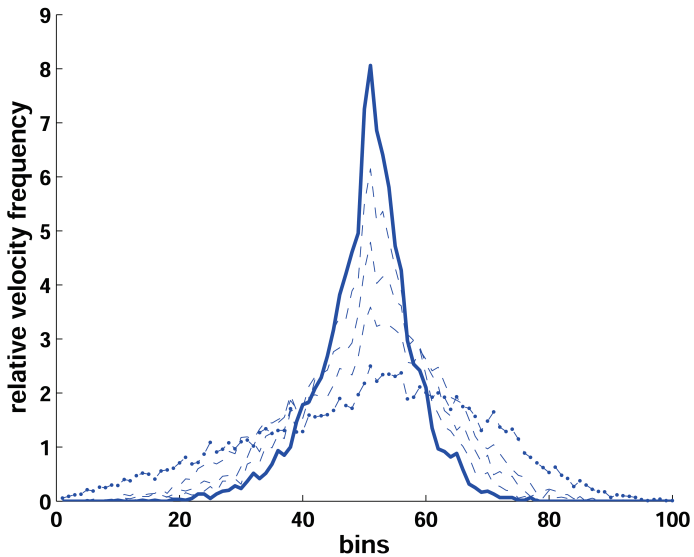
Crucial assumptions: the turning kernel $\mu(y, s)$ depends on the state y, where

$$\frac{dS}{dt} = \Delta S + \phi(S)$$

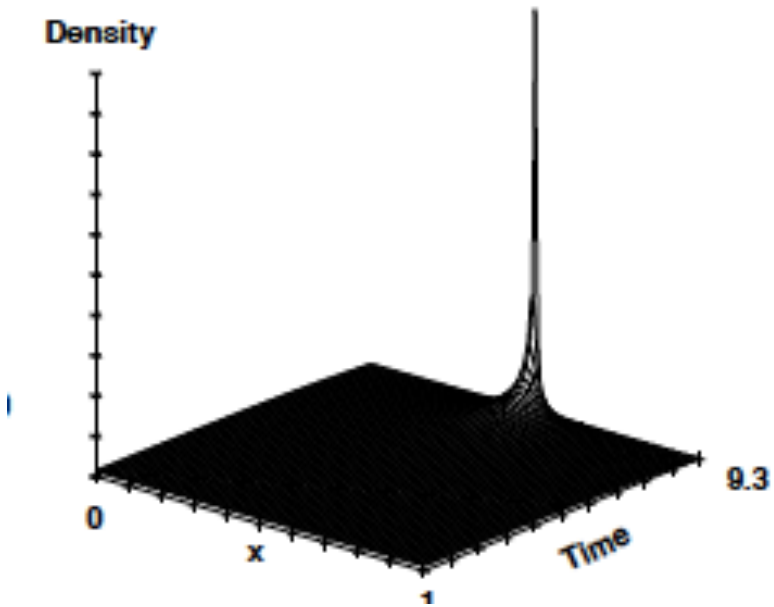
is the internal state equation for the system.

Blow-up of solutions in macroscopic limit

Evolution of $\rho(v,t)$ in 1-d model of grain growth model



Evolution of concentration in a 1-d velocity jump model (Othmer, Stevens):



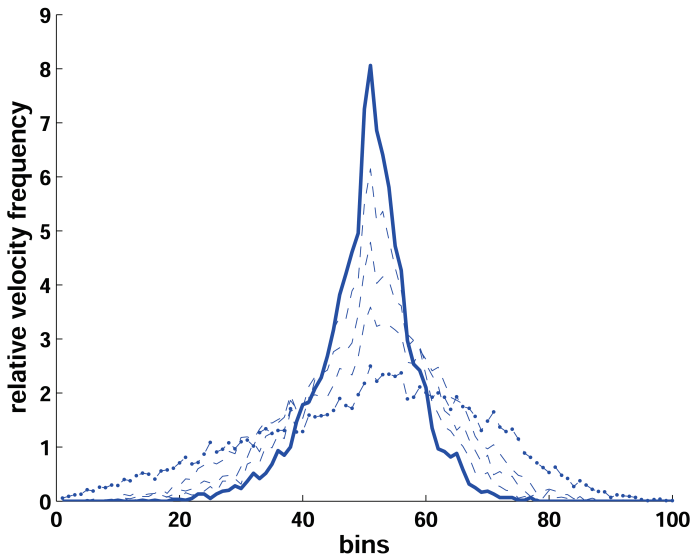
This model belongs to the same family of kinetic equations (see Othmer et al 1988):

$$\partial_t f(x,v,t) + v \cdot \nabla_x f = \int_V (T[S]f' - T^*[S]f)dv'$$
$$\partial_t S = D_S \Delta S + \phi(S)$$

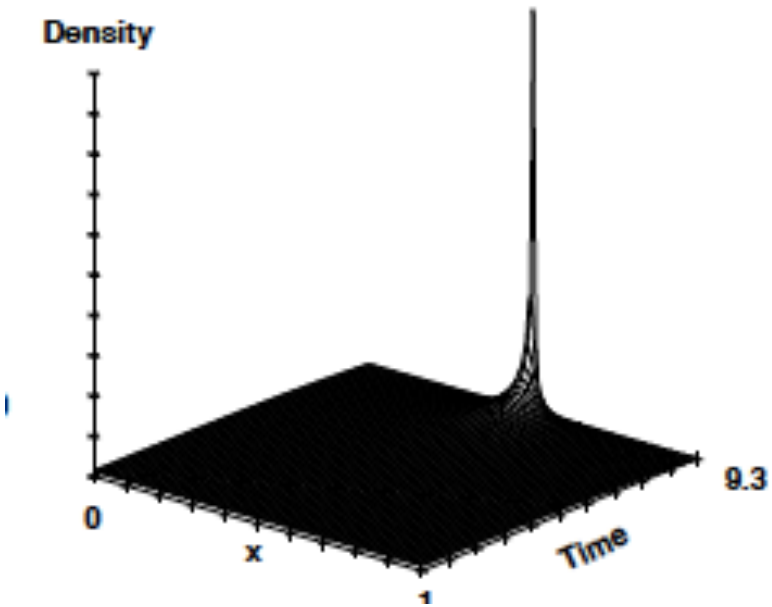
where S is the evolving "environment" variable accounting for the correlations in the velocity space. First derived for a 1d grain growth model in [Barmak, Emelianenko, Golovaty, Kinderlerer, Taasan 2008].

Blow-up of solutions in macroscopic limit

Evolution of $\rho(v,t)$ in 1-d model of grain growth model



Evolution of concentration in a 1-d velocity jump model (Othmer, Stevens):



This model belongs to the same family of kinetic equations (see Othmer et al 1988):

$$\partial_t f(x,v,t) + v \cdot \nabla_x f = \int_V (T[S]f' - T^*[S]f)dv'$$
$$\partial_t S = D_S \Delta S + \phi(S)$$

where S is the evolving "environment" variable accounting for the correlations in the velocity space. First derived for a 1d grain growth model in [Barmak, Emelianenko, Golovaty, Kinderlerer, Taasan 2008].

Implications for microstructure modeling

- This connections allow to explain the blow-up of solutions
- Passing to diffusion limit yields the famous Keller-Segel macroscopic equations:

$$\partial_t f = \nabla \cdot (D \nabla f - \chi \nabla f)$$
$$\partial_t S = D_S \Delta S + \phi(S)$$

  where $f(l, t)$ is the distribution of the GB length.
- The evolution of the S variable can be both local or nonlocal
- Other distributions like area, number of sides statistics, can be modeled via same formalism.
- Higher dimensional generalizations are possible but more technically involved.

Summary for Problem 2

- Numerical simulations have to be subject to accurate numerical analysis in order to guarantee adequate performance
- Vertex codes save time, but can produce unreliable dynamics and/or mesoscopic behavior if not performed properly
- It is possible to minimize the amount of random choices in vertex code simulations, based on the physical principles
- TJ velocity jump process can be modeled via coupled integro-differential equations, with the carefully chosen set of parameters.
- Solutions of the mesoscopic equations derived in this context can be smooth, or blow up in finite time. Further analysis is necessary to make the theory complete.

THANKS!