MDDP-suite

Multiscale Dislocation Dynamics Plasticity Method

MDDP: micro3d   fea3d   ht3d

For FCC and BCC Single Crystals

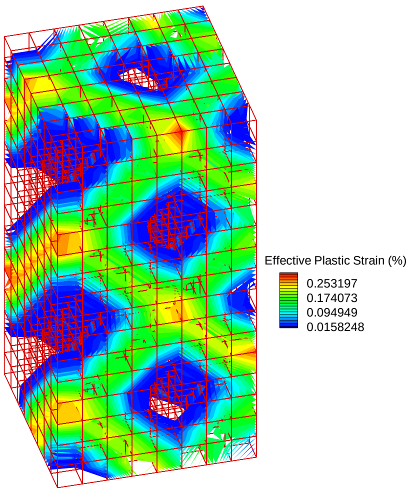
Version 02

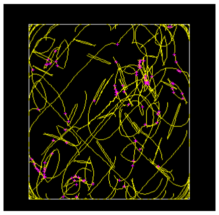
Developer

Hussein M. Zbib

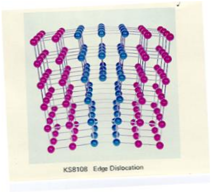
User's Manual for internal Use Only

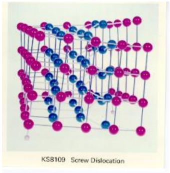
Washington State University
Pullman, WA 99164





1993-2002





Contents

This work has been supported by the US National Science Foundation, Lawrence Livermore National Laboratory, and the Pacific Northwest National Laboratory

Contributed to this development:
Professor John P. Hirth
and
Graduate and Post-Doc Students:
Moon Rhee, Tariq Khraishi, Hasan Yasin, Shafique Khan, Masato Hiratani

List of Figures

List of Tables

1. Description of the multiscale dislocation dynamics plasticity "*MDDP02*" and the Discrete Dislocation Dynamics Model "*micro3d*"

micro3d:

The physical model is based on a large number of discrete dislocation segments situated into a computational cell representing a *continuum* of 3D single crystal, Fig. 1. The cell is further divided into sub-cells, or link cells, for more efficient calculation of long range interactions (Hirth et al., 1996 and Zbib et al., 1997). Each cell contains a number of dislocation loops and lines of arbitrary shapes lying on slip systems, as illustrated in Fig.1. The model has been developed for both fcc and bcc single crystals. Typical dislocation density ranges from very low, $10^8\ \mathrm{m}^{-2}$ (annealed) to very high $10^{15}\ \mathrm{m}^{-2}$ (highly strained). In *micro3d,* arbitrarily curved dislocations are decomposed into piecewise continuous arrays of *mixed segments* in a *continuum crystal*. Depending on the local curvature, the spacing between two dislocation nodes $\Delta\ell$ varies from 3b to 100b (adaptive re-meshing) (Zbib et al., 1997), resulting into N segments. Table 1 lists typical values for N. This implies that large computational requirements are, indeed, needed to solve this problem, especially since the interaction problem is of the order N^2.

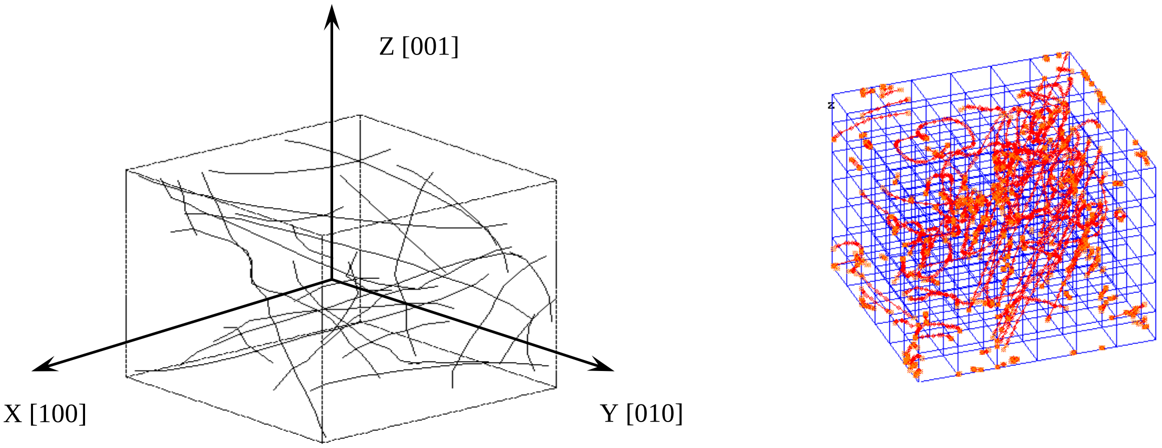


Figure 1: a) Computational cell of a bcc single crystal, b) random distribution of dislocation lines in bcc crystal with initial density of 9.91x10^{11}m^{-2}

NOTES:

a) **Coordinate System:** Reference global coordinate system is located at the center of the cell with the axes coinciding with the crystal axes. "Rotated" axes can be defined in the file "*data*"

b) **Non-dimensional space:** Space is non-dimensionalized by the magnitude of the Burgers vector, b;

$$L=L/b$$

Table 1: Typical number of dislocation segments in relation to dislocation density and cell size

Cell size, μm	**Dislocation density, m^{-2}**	**Number of segments, *N*** **($\Delta\ell = 100b - 1000b$)**
10	$10^{10} - 10^{15}$	$10^{2} - 10^{6}$
50	$10^{10} - 10^{15}$	$10^{3} - 10^{7}$

a. Computation of The Dislocation Stress Field

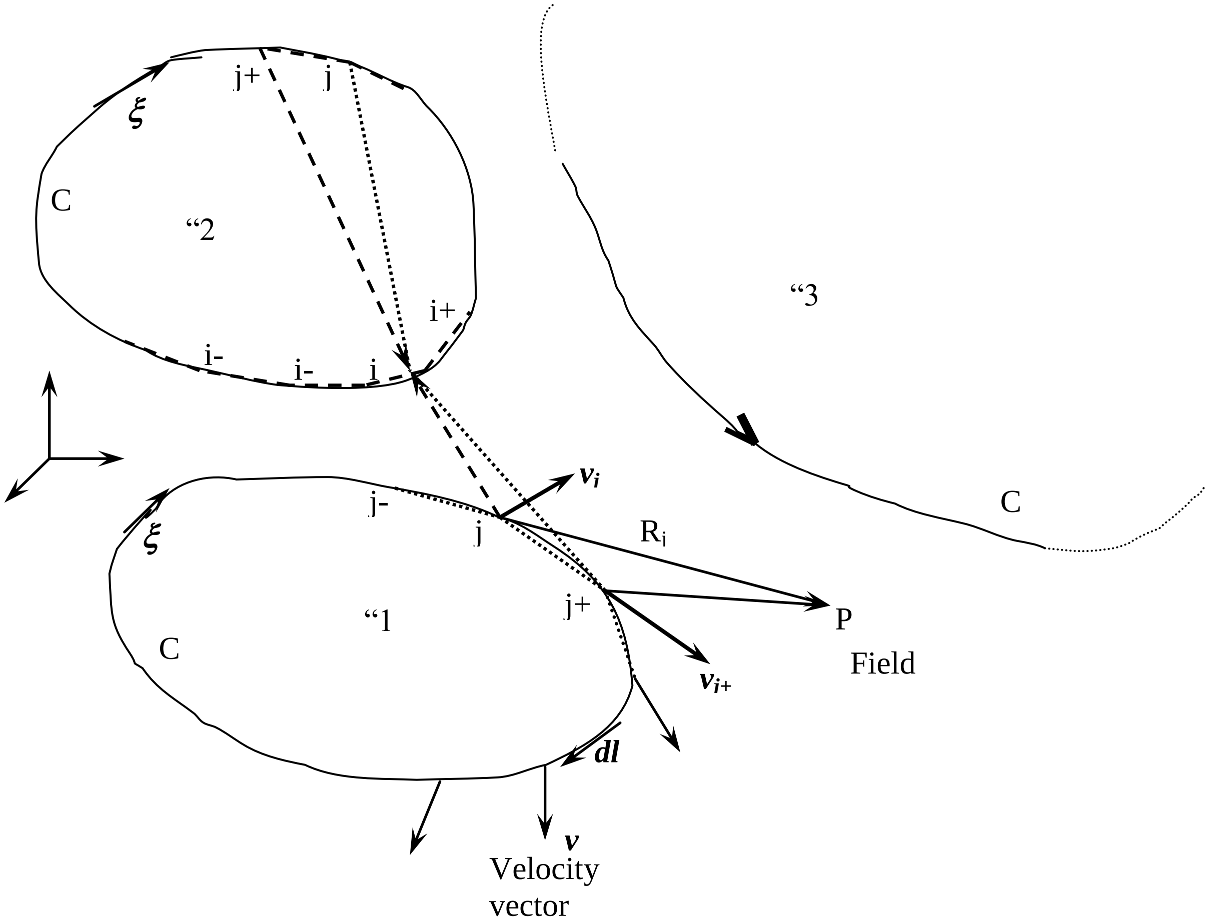


Figure 2: Dislocations curves: Meshing and dislocation nodes

$$\sigma_{\alpha\beta}(\,\boldsymbol{p}\,) = \sum_{\boldsymbol{allLoops}} \sum_{\boldsymbol{j}=1}^{\boldsymbol{n}-1} \{ -\frac{G}{8\pi} \int_{\boldsymbol{j}}^{\boldsymbol{j}+1} b_m \in_{im\alpha} \frac{\partial}{\partial x'_i} \nabla'^2 R\, dx'_\beta - \frac{G}{8\pi} \int_{\boldsymbol{j}}^{\boldsymbol{j}+1} b_m \in_{im\beta} \frac{\partial}{\partial x'_i} \nabla'^2 R\, dx'_\alpha$$
$$- \frac{G}{4\pi(1-\nu)} \int_{\boldsymbol{j}}^{\boldsymbol{j}+1} b_m \in_{imk} \left(\frac{\partial^3 R}{\partial x'_i \partial x'_\alpha \partial x'_\beta} - \delta_{\alpha\beta} \frac{\partial}{\partial x'_i} \nabla'^2 R \right) dx'_k \} \qquad (1)$$

Yields:

$$\boldsymbol{F_i} = \sum_{\boldsymbol{j}=1}^{\boldsymbol{N}-1} (\,\boldsymbol{\sigma}^{\boldsymbol{D}}_{\boldsymbol{j},\boldsymbol{j}+1} + \boldsymbol{\sigma}^{\boldsymbol{a}}\,).\boldsymbol{b_i} \times \boldsymbol{\xi_i} + \boldsymbol{F_{i-self}} \qquad (2)$$

2. Problem Statement

MDDP02 combines "*micro3d*" with finite element codes "*fea3d"* (solid mechanics) and *ht3d* (heat transfer). It has been constructed within a generalized enough framework so that many classes of dislocation problems can be investigated, including:

a) Simple dislocation mechanisms, such as Frank-Read sources.
b) Stability of dislocation cell structures, such as cell walls and dislocation boundaries.
c) Evolution of random distribution of dislocations.
d) Interaction among dislocations and point defects and particles, SFT's, loops, etc.

In this respect, users subroutines to include effects of other defects can be easily incorporated into *micro3d*, e.g. a subroutine to include the effect of prismatic dislocation loops to deal with the radiation-induced hardening problem, a subroutine to deal with dislocations interacting with microcracks, etc.

Therefore, the main issue is to predict the spatio-temporal evolution of the dynamical and self-organizing system consisting of *N* dislocation segments, and the manner in which they interact with each other and other defects and surfaces to determine the overall strength of the metal under various loading conditions. The fundamental aspects of the model are built from the basic physical laws that govern:

a) the mobility of an individual dislocation,
b) short range interactions between two dislocations on core level, and
c) long-range interactions associated with elastic distortions.

The main governing equation for the dynamics of each dislocation segment is given by (Hirth and Lothe, 1982; Indenbom and Lothe, 1992; Hirth, Zbib and Lothe, 1997)

$$m_i^* \dot{\boldsymbol{v}}_i + \frac{1}{M_i(T,p)} \dot{\boldsymbol{v}}_i = \left[\sum_{j=1}^{N-1} (\boldsymbol{\sigma}_{j,j+1}^{D} + \boldsymbol{\sigma}^{a}).\boldsymbol{b}_i \times \boldsymbol{\xi}_i + \boldsymbol{F}_{i-self} \right]_{glide-component} \qquad (3)$$

Here $\boldsymbol{F}_i(\boldsymbol{v})$ is the inertial force, $\boldsymbol{v}_i$ is the dislocation segment velocity, M_g is the mobility, $\boldsymbol{F}^{a}$ is the force produced by applied stresses, and $\boldsymbol{F}_i^{int}$ is the internal force arising from interactions with other defects and dislocations and from the Peierls barrier if present (see Appendix A). Calculation of the long range interaction is most expensive (order N^2). Therefore, we developed a method (superdislocation method) to reduce the order of interaction (to *NlogN*) with high accuracy (Hirth et al., 1996; Zbib et al., 1997) (the analog of the 2D multipolar expansion method).

Determination of the mobility and interaction forces (long range elastic stress fields, and short range) constitutes the core of the model. Generally, M_g is a function of the angle between the Burgers vector and the dislocation line sense, especially at low temperatures. In bcc single crystals, at low temperatures a pure screw dislocation has a rather complex three-dimensional core structure, resulting in a high Peirels stress which is overcome by stress-assisted thermal activation (Hirth and Lothe, 1982). This leads to a relatively low mobility for screw dislocations while the mobility of mixed dislocations is very high (Urabe and Weertman, 1975). The kinetics of a screw dislocation is characterized by the mechanism of the succession of kink nucleation and lateral double kink migration, which are edge dislocations. This theory leads to a temperature-dependent mobility with activation enthalpy associated with kink nucleation. Basic relations are given in Hirth and Lothe (1982) and constitute the core of the model for mobility.

3. *micro3d – fea3d : MDDP02*

The Coupled continuum mechanics – discrete dislocation dynamics approach (multi-scale approach)

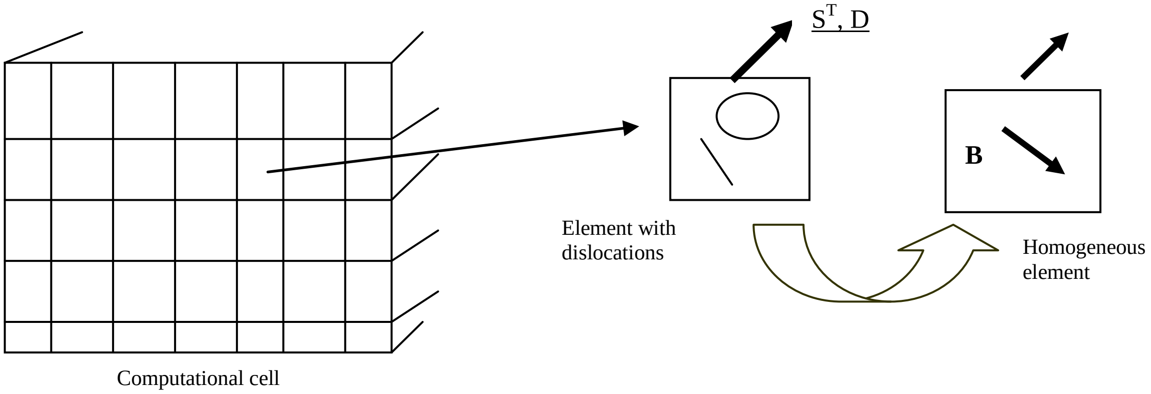


Figure 3: Quasi-continuum finite elements with discrete dislocations
The total stress $\boldsymbol{S^T}$ in the *RVE* element arises from: 1) applied loads on the surface of the computational cell, and 2) stress $\boldsymbol{S^D}$ from the elastic distortion of the dislocations in the whole solid, which also include dislocation image stresses arising from free boundaries (if any).

Here we give a brief summary of the continuum mechanics framework that we have developed and the manner in which we coupled it with the DD code. Generally, the basic governing equations of the material response in continuum mechanics are developed based on a *representative volume element* (*RVE)* over which the deformation field is assumed to be homogeneous. Typically, in this approach the effect of internal defects, such as dislocations, voids, microcracks etc., on material behavior and the manner they influence material properties is modeled through a set of internal variables and corresponding phenomenological evolution equations. The material response is measured in terms a macroscopic strain rate tensor $\boldsymbol{D}$ and its relation to the Cauchy stress tensor $\boldsymbol{S}$. Furthermore, for elasto-viscoplastic behavior, the strain tensor $\boldsymbol{D}$ is decomposed into the sum of an elastic and plastic, $\boldsymbol{D^e}$ and $\boldsymbol{D^p}$, respectively**.** For most metals the elastic response is linear and is expressed by the incremental form of Hooke's law for large deformation and material rotation, i.e. $\overset{o}{\boldsymbol{S}} = [\boldsymbol{L^e}]\, \boldsymbol{D^e}$, $\overset{o}{\boldsymbol{S}} = \dot{\boldsymbol{S}} - \omega\ \boldsymbol{S} + \boldsymbol{S}\, \omega$, $\omega = \boldsymbol{W} - \boldsymbol{W^p}$, where $\boldsymbol{L^e}$ is a fourth order tensor, $\boldsymbol{\omega}$ is the spin of the substructure and is given as the difference between the material spin $\boldsymbol{W}$ and plastic spin $\boldsymbol{W^p}$. **The main issue here is to evaluate $\boldsymbol{D^p}$ and $\boldsymbol{W^p}$ and relate them to the underlying defect structure, mainly dislocations.** Independent of the nature of plastic strain tensor, we can use standard variational principal and cast the FE problem, for quasi-static case, into the standard form (after re-writing the equations in total form as opposed to the incremental form):

$$[\boldsymbol{M}]\{\ddot{\boldsymbol{U}}\}+[\boldsymbol{C}]\{\dot{\boldsymbol{U}}\}+[\boldsymbol{K}]\{\boldsymbol{U}\}=\{\boldsymbol{f}^{a}\}+\{\boldsymbol{f}^{B}\}+\{\boldsymbol{f}^{\infty}\}+\{\boldsymbol{f}^{P}\} \qquad (2)$$

where $[M]=\int\limits_V \rho[N]^T[N]dv$ is the mass matrix, $[\boldsymbol{K}]=\int\limits_V[\boldsymbol{B}]^T[\boldsymbol{C}^e][\boldsymbol{B}]dv$ is the stiffness matrix, $\{\boldsymbol{f}^a\}=\int\limits_s \boldsymbol{t}^a[\boldsymbol{N}]ds$ is the applied force vector, $\{\boldsymbol{f}^\infty\}=\int\limits_s \boldsymbol{t}^\infty[\boldsymbol{N}]ds$ is the force vector from dislocation image stresses, $\{\boldsymbol{f}^B\}=\int\limits_v \boldsymbol{S}^D[\boldsymbol{B}]\boldsymbol{dv}$ is the body force vector from dislocations long-range interaction and $\{\boldsymbol{f}^P\}=\int\limits_v[\boldsymbol{C}^e]\boldsymbol{\varepsilon}^P[\boldsymbol{B}]\boldsymbol{dv}$ is force vector from plastic strain caused by dislocations, with [**N**] being the shape function vector, [**B**]=grad[**N**], {u}=[N]{U}, and $\boldsymbol{\varepsilon}=[B]\{U\}$. Dislocations are sorted out in each element and they contribute to the plastic strain based on equation (13).

Numerical solution- explicit integration: Both the DD system of equations and the dynamic finite element model are solved using a forward explicit integration scheme. This scheme is chosen since the time step in the DD analysis (for high strain rates) is of the same order of magnitude of the time required for stable explicit *FE* dynamic analysis (*FEA*). In this analysis, the critical time t_c and the time step for both the *DD* and the *FEA*, which yield a stable solution, are given by

$$\boldsymbol{t_c}=\frac{\ell_{\boldsymbol{c}}}{\boldsymbol{C}_\ell}, \quad \boldsymbol{\delta t}=\frac{\boldsymbol{t_c}}{\boldsymbol{20}}$$

where $\ell_{\boldsymbol{c}}$ is the characteristic length scale which is the shortest dimension in the finite element mesh.

Coupling with the discrete-dislocation dynamics code *micro3d: MDDP* The *"assumed*" constitutive nature of the plastic deformation tensor $\boldsymbol{D}^{\boldsymbol{p}}$ and flow stress and their dependence upon internal variables and gradients of internal variables is very critical, since they dictate, among other things, the length scale of the problem and the phenomena that the model can capture. However, the discrete dislocation dynamics model (*micro3d*) provides the most rigorous and physically based approach for computing the plastic strain and strain hardening in metals through an explicit evaluation of the motion and evolution of all individual discrete dislocations in the crystal. Therefore, *fea3d* is made as an integral part of *micro3d* as depicted below (Fig.4).

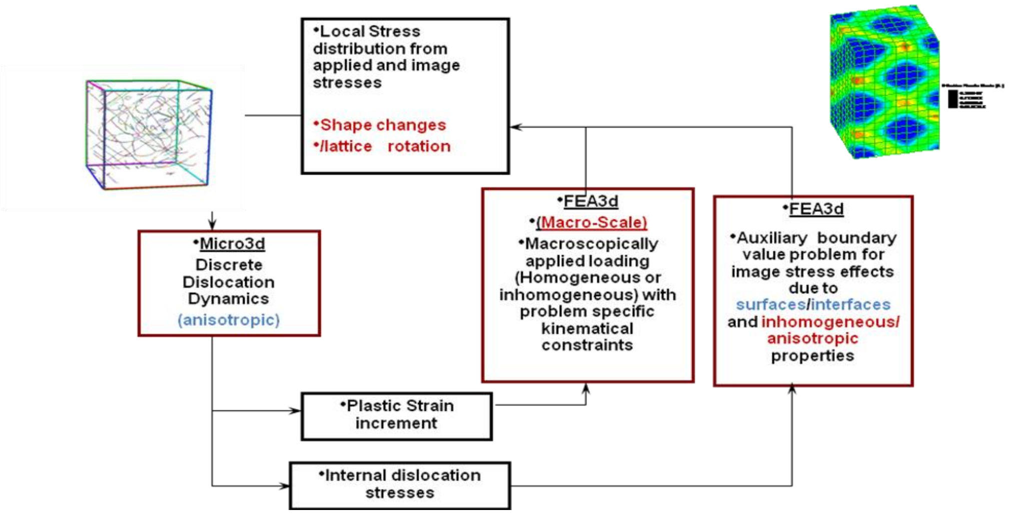

•Local Stress distribution from applied and image stresses
•Shape changes
•/lattice rotation
•Micro3d
Discrete Dislocation Dynamics
(anisotropic)
•FEA3d
•(Macro-Scale)
•Macroscopically applied loading (Homogeneous or inhomogeneous) with problem specific kinematical constraints
•FEA3d
•Auxiliary boundary value problem for image stress effects due to surfaces/interfaces and inhomogeneous/ anisotropic properties
•Plastic Strain increment
•Internal dislocation stresses

Figure 4:Coupling *micro3d* with *fea3d: MDDP02*

4. Computational Cell and Boundary Conditions

The computational cell could be a representation of one of the two:

1) A representative cell in an infinite domain.
2) The whole test specimen: Finite domain.

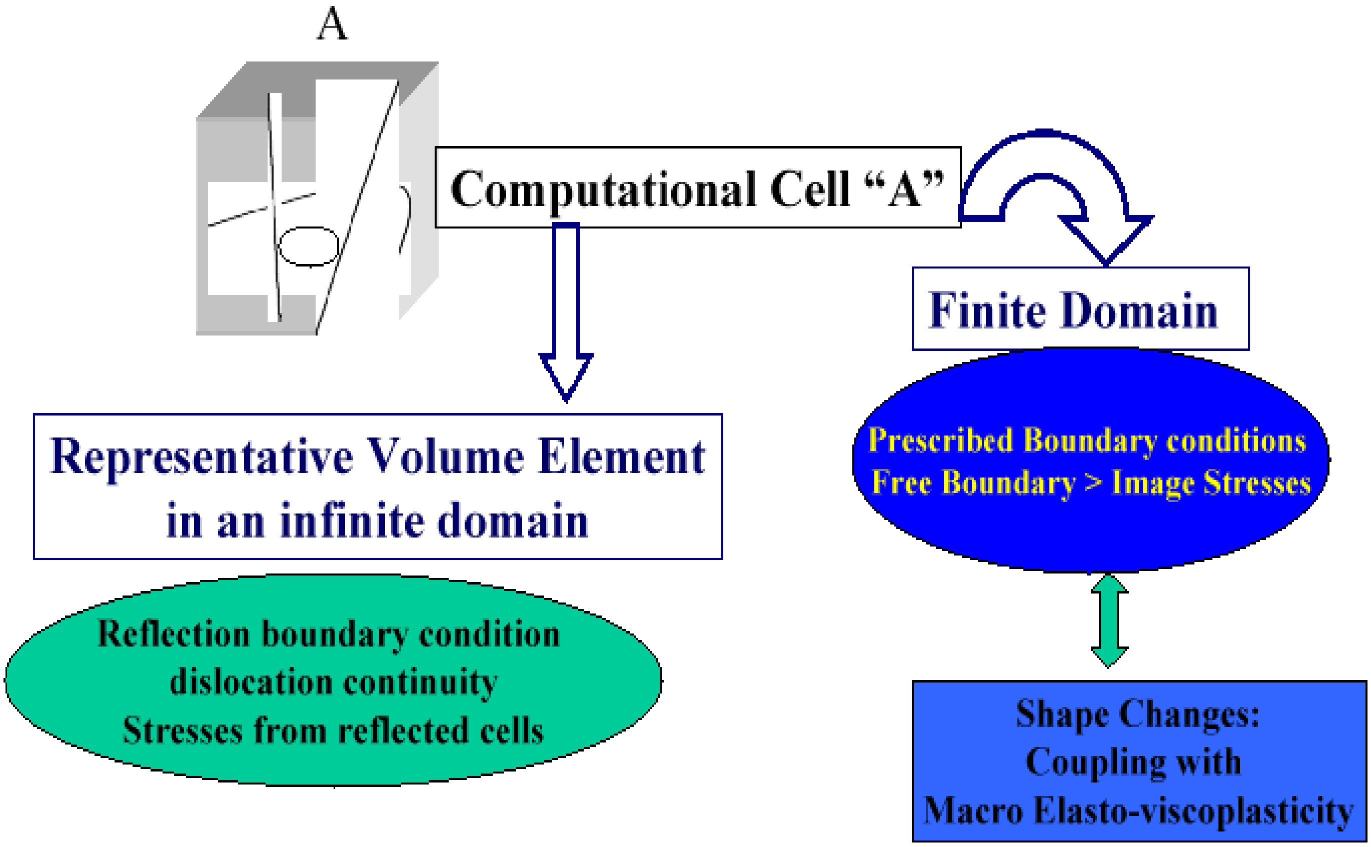

A
Computational Cell "A"
Finite Domain
Representative Volume Element in an infinite domain
Prescribed Boundary conditions
Free Boundary > Image Stresses
Reflection boundary condition
dislocation continuity
Stresses from reflected cells
Shape Changes:
Coupling with
Macro Elasto-viscoplasticity

Figure 5: Computational cell

a. Infinite Domain

Dislocations in computational cell are reflected across the cell boundaries:

- Maintains continuity of dislocation lines
- Reflected cells
- Multiple-time step: DT for updating far stress field > dt

Computation of the *1/r* Stress Field:
Dislocations in CC are divided into "M" sub-cells. Then the computational strategy is as follows:

- Direct interaction with immediate neighbor **subcells**
- stress from far dislocations is computed at center of **subcell**

- Stresses from reflected cells is computed using the “Superdislocation Method” $N^2 \rightarrow NLog(N)$

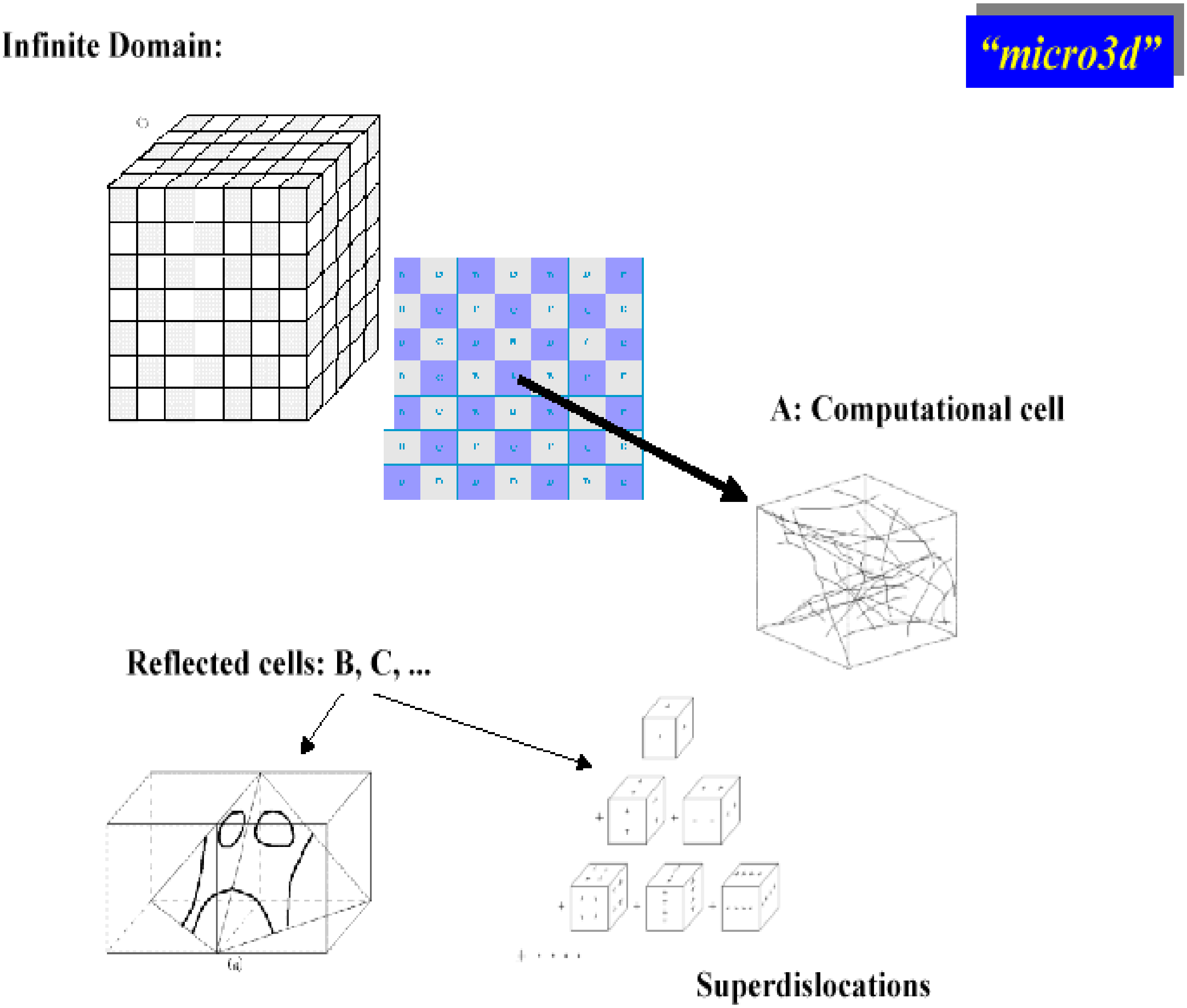


Figure 6: Infinite Domain: Computational cell and reflected cells
The dislocations in the reflected cells are combined into superdislocations for long range interaction.

b. Finite Domain: Coupled FEA –DD

(Image stresses)

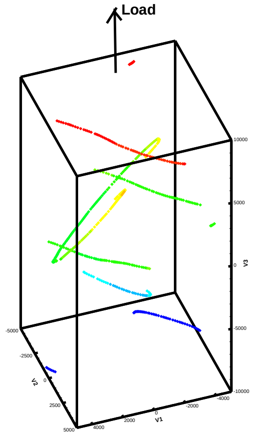


Figure 7: Finite domain: Free boundaries
The computational cell in this case has free boundaries

The simulation cell may represent:

- Test specimen (micrometer size), Thin film, etc.

Boundary conditions:

- Free and/or rigid surfaces:
- Free surfaces: Zero traction, image stresses from dislocations within the cells
- Rigid surfaces: Zero displacement(relative), image displacements (Heterogeneous fields)
- Applied stresses:
  Tension, compression, nano-indentor, etc.
  (homogeneous or heterogeneous)

Finite Element Framework (*fea3d*):
Stress at any point in the cell =
stresses from dislocations and internal defects
+ applied stress
+ stresses from image forces

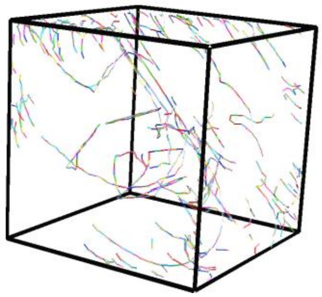
- The Computational cell is divided into finite elements.
- Stress field arising from Image stresses is computed using the finite element method (auxiliary problem). Boundary value problem of a linear elastic continuum

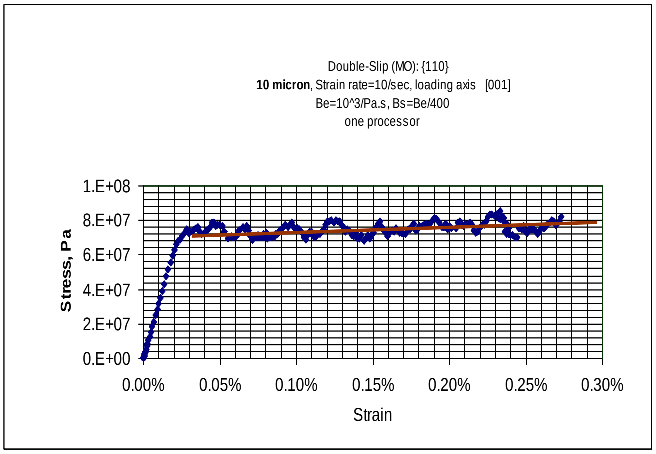
Shape changes (finite plastic deformation):
Coupling of DD with FE viscoplastic formulation.

5. The Contents of the Computational Cell

a. Dislocations

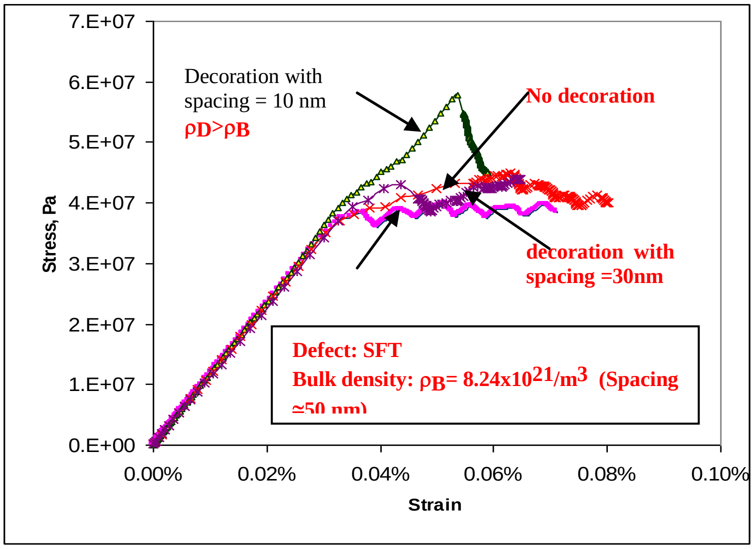
Frank-Read Sources, Random distribution of dislocation curves, etc.

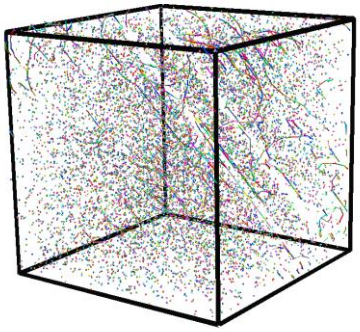




b. Point defects

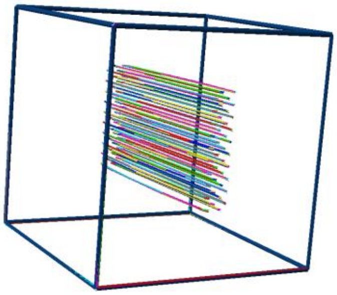
Stacking-Fault Tetrahedron
Frank-sessile loops
(millions of defects)

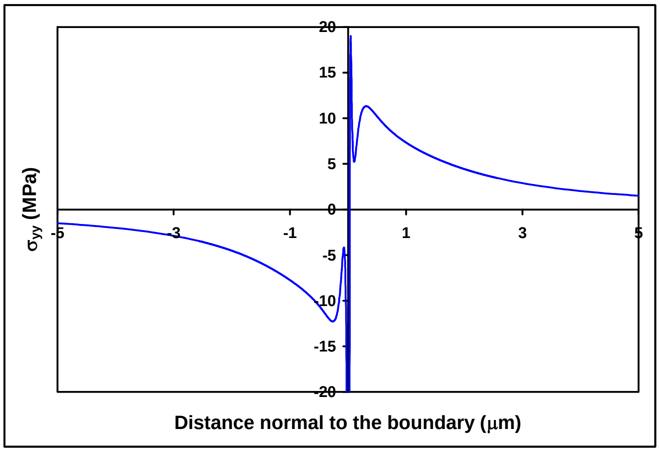




c. Dislocation boundaries

Cell walls





***<u>Or Combination of all of the above</u>**

d. Pre-processors for data generation

a) GendataBCC.F
b) GendataFCC.F

6. Description of Dislocation Geometry and Constraints

Each dislocation node (defining a segment) is described by the following:

a) Nodal coordinates
b) Slip plane
c) Burgers vectors index (for cross-slip Data)
d) Magnitude of Burgers vector
e) Nodal constraints
f) Junction and Jog index

a. Discretization and Description of Nodal Coordinates

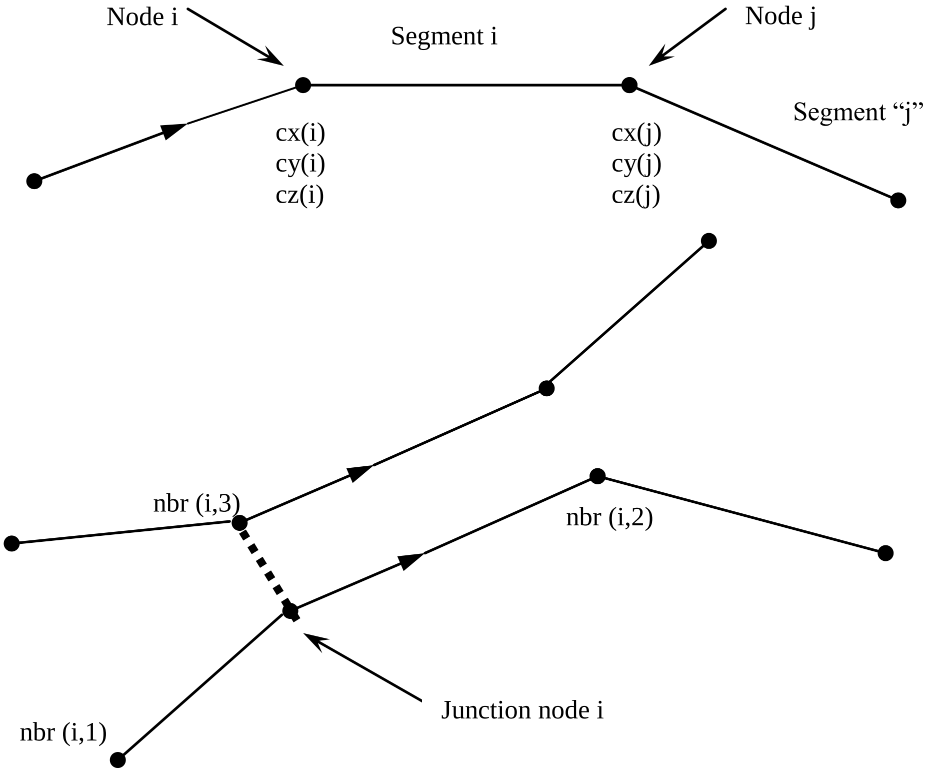


Figure 8: Junction node

cx(i), cy(i), cz(i)	Nodal coordinates
bx(i), by(i), bz(i)	Burgers vector in Cartesian coordinates
nbr(i,1)	backward neighbor of node "i" (relative to line sense)
nbr(i,2)	forward neighbor of node "i" (relative to line sense)
nbr(i,3)	If "i" is a junction node then it has a third neighbor nbr(i,3) =k and segment i-k is the junction, otherwise nbr(i,3)=0

b. Type of Nodal Constraints and Corresponding Index icn(i)

Table 2: Nodal constraints

Constraint Type	**icn(i) index**
Free node	0
Surface node: On yz plane On xz plane On xy plane	 1 2 3
Node at corner	4
Cross-slip node	5
Pinned node	7
Jog node	9
Junction node	10

Table 3: Vectors assigned to each dislocation node

Main Vector	**For Each Node**
cx(i), cy(i), cz(i)	x, y, z coordinates
nbr(i,1), nbr(i,2), nbr(i,3)	defines neighbors
glbx(i), glby(i), glbz(i)	x, y, z components of Burgers vector
bjuncx(i), bjuncy(i),bjuncz(i)	x, y, z components of Burgers vector of junction
iplane(i)	index defining slip plane
ixbtype(i)	index defining Burgers vector
ixpltyp(i,j)	index defining common planes “j” for each Burgers vector “i” (e.g. for fcc, j=1,2 ixpltyp(1,1)=3 ixpltyp(1,2)=4)
icn(i)	constraint index defining motion of node
glveL(i)	glide velocity of segments
pchx(i), pchy(i), pchz(i)	x, y, z components of Peach Koehler force
fglide(i)	glide force
jndx(i)	junction index = 0 ⇒ node not a junction node, or = junction number
jogndx(i)	jog index = 0 ⇒ node not a jog or = jog number

c. Slip Planes and Corresponding Indices

BCC System

The slip planes {110} and {112} are considered. There are 18 possible slip planes, and four Burgers vector. See Appendix A for all possible slip systems. A screw dislocation for a given Burgers vector could cross-slip on any of six planes defined by the index ixpltyp(i,j).

Table 4: Indices for slips planes and Burgers vectors (bcc)

Slip Planes

Index iplane(i)	Plane
1	$(01\bar{1})$
2	(011)
3	$(10\bar{1})$
4	(101)
5	(110)
6	$(1\bar{1}0)$
7	$(2\bar{1}1)$
8	$(21\bar{1})$
9	$(2\bar{1}\bar{1})$
10	(211)
11	$(\bar{1}\bar{1}\bar{2})$
12	$(11\bar{2})$
13	$(1\bar{1}\bar{2})$
14	$(\bar{1}1\bar{2})$
15	$(1\bar{2}1)$
16	$(\bar{1}21)$
17	$(\bar{1}\bar{2}1)$
18	$(\bar{1}\bar{2}\bar{1})$

Data for cross-slip in BCC for planes {110} & {112}

Burgers Vector Index ixbtyp(i)	Burgers vector	cross-slip index ixpltyp(i,j)	Slip System	
1	$[111]$	1	$[111]\ (01\bar{1})$	1
		3	$[111]\ (10\bar{1})$	2
		6	$[111]\ (1\bar{1}0)$	3
		9	$[111]\ (2\bar{1}\bar{1})$	4
		12	$[111]\ (11\bar{2})$	5
		15	$[111]\ (1\bar{2}1)$	6
2	$[\bar{1}1\bar{1}]$	2	$[\bar{1}1\bar{1}]\ (011)$	7
		3	$[\bar{1}1\bar{1}]\ (10\bar{1})$	8
		5	$[\bar{1}1\bar{1}]\ (110)$	9
		8	$[\bar{1}1\bar{1}]\ (21\bar{1})$	10
		16	$[\bar{1}1\bar{1}]\ (\bar{1}21)$	11
		13	$[\bar{1}1\bar{1}]\ (1\bar{1}\bar{2})$	12
3	$[\bar{1}\bar{1}1]$	2	$[\bar{1}\bar{1}1]\ (011)$	13
		4	$[\bar{1}\bar{1}1]\ (101)$	14
		6	$[\bar{1}\bar{1}1]\ (1\bar{1}0)$	15
		7	$[\bar{1}\bar{1}1]\ (2\bar{1}1)$	16
		11	$[\bar{1}\bar{1}1]\ (\bar{1}\bar{1}\bar{2})$	17
		18	$[\bar{1}\bar{1}1]\ (\bar{1}\bar{2}\bar{1})$	18
4	$[\bar{1}11]$	1	$[\bar{1}11]\ (01\bar{1})$	19
		4	$[\bar{1}11]\ (101)$	20
		5	$[\bar{1}11]\ (110)$	21
		10	$[\bar{1}11]\ (211)$	22
		14	$[\bar{1}11]\ (\bar{1}1\bar{2})$	23
		17	$[\bar{1}11]\ (\bar{1}\bar{2}1)$	24

FCC System

The slip planes {111} are considered. There are 4 possible slip planes, and six Burgers vector. See Appendix A for all possible slip systems. A screw dislocation for a given Burgers vector could cross-slip on any of two planes defined by the index ixpltyp(i,j).

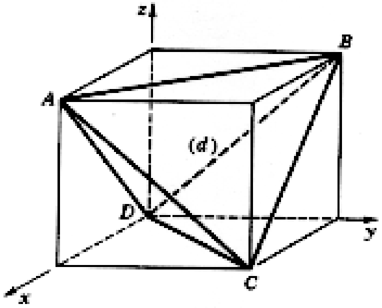


Table 5: Indices for slips planes and Burgers vectors (fcc)

Slip planes

Index Iplane(i)	Plane
α 1	$(1\bar{1}1)$
β 2	$(\bar{1}11)$
γ 3	$(\bar{1}\,\bar{1}\,1)$
δ 4	(111)
(Using Thompson's Tetrahedron notation)	

Data for cross-slip in FCC for planes {111}

Burgers Vector Index ixbtyp(i)	Burgers vector	cross-slip index ixpltyp(i,j)
AB 1	$[\bar{1}10]$	3 4
AC 2	$[01\bar{1}]$	2 4
AD 3	$[\bar{1}0\bar{1}]$	2 3
BC 4	$[10\bar{1}]$	1 4
BD 5	$[0\bar{1}\bar{1}]$	1 3
CD 6	$[\bar{1}\bar{1}0]$	1 2

7. Stacking-fault Tetrahedron

a. Stair Rods Dislocations

$$b_{\delta\beta} = b_{\delta A} + b_{A\beta} = \frac{a}{6}[1\bar{2}1] + \frac{a}{6}[\bar{1}1\bar{2}] = \frac{a}{6}\left[0\bar{1}\,\bar{1}\right]$$

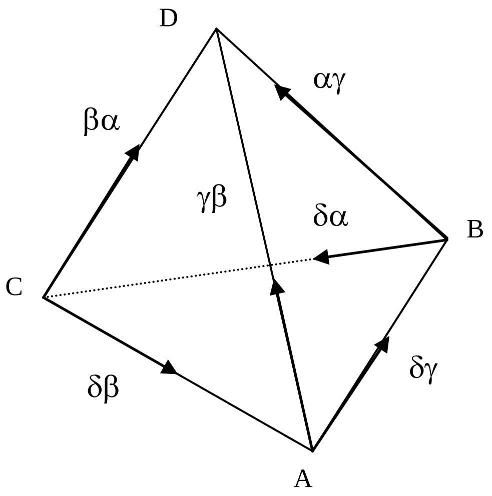
$$b_{\delta\gamma} = b_{\delta B} + b_{B\gamma} = \frac{a}{6}[\bar{2}11] + \frac{a}{6}[1\bar{2}\,\bar{1}] = \frac{a}{6}\left[\bar{1}\,\bar{1}0\right]$$

$$b_{\delta\alpha} = b_{\delta C} + b_{C\alpha} = \frac{a}{6}[11\bar{2}] + \frac{a}{6}[\bar{2}\,\bar{1}1] = \frac{a}{6}\left[\bar{1}0\bar{1}\right]$$

$$b_{\beta\alpha} = -b_{\delta\beta} + b_{\delta\alpha} = \frac{a}{6}\left[\bar{1}10\right]$$

$$b_{\gamma\beta} = -b_{\delta\gamma} + b_{\delta\beta} = \frac{a}{6}\left[10\bar{1}\right]$$

$$b_{\alpha\gamma} = -b_{\delta\alpha} + b_{\delta\gamma} = \frac{a}{6}\left[0\bar{1}1\right]$$



b. Shockley partials

$$A\beta : \frac{a}{6}\left[\bar{1}1\bar{2}\right] \qquad C\beta : \frac{a}{6}\left[\bar{1}\,\bar{2}1\right]$$

$$A\beta : \frac{a}{6}\left[\bar{1}1\bar{2}\right] \qquad C\alpha : \frac{a}{6}\left[\bar{2}\,\bar{1}1\right]$$

$$D\alpha : \frac{a}{6}\left[121\right] \qquad B\alpha : \frac{a}{6}\left[1\bar{1}\,\bar{2}\right]$$

$$A\gamma : \frac{a}{6}\left[\bar{2}1\bar{1}\right] \qquad B\gamma : \frac{a}{6}\left[1\bar{2}\,\bar{1}\right]$$

$$D\gamma : \frac{a}{6}\left[112\right] \qquad C\delta : \frac{a}{6}\left[\bar{1}\,\bar{1}2\right]$$

$$B\delta : \frac{a}{6}\left[2\bar{1}\,\bar{1}\right] \qquad A\delta : \frac{a}{6}\left[\bar{1}2\bar{1}\right]$$

$a = b\sqrt{2}$

b: magnitude of perfect <*110*> dislocation

8. Program Description and Data Files

a. Main Model Parameters

a.1 Physical parameters

Elastic properties:	E, ν
Mobility:	B_{edge}, B_{Screw},
Jog strength-critical angle:	θ_j
Self force - core size:	
Junction strength - core size:	

a.2 Numerical Parameters

Cell size

Segment length (min and max)

Number of cells (for infinite domain)

Number of sub-cells (elements for FE)

Initial time step stress control Δt

(for constant strain rate: variable time step is controlled by the amount of plastic strain increment)

Initial time step for DD δt $(\Delta t = \sum \delta t)$

Max flight distance: (variable time step δt)

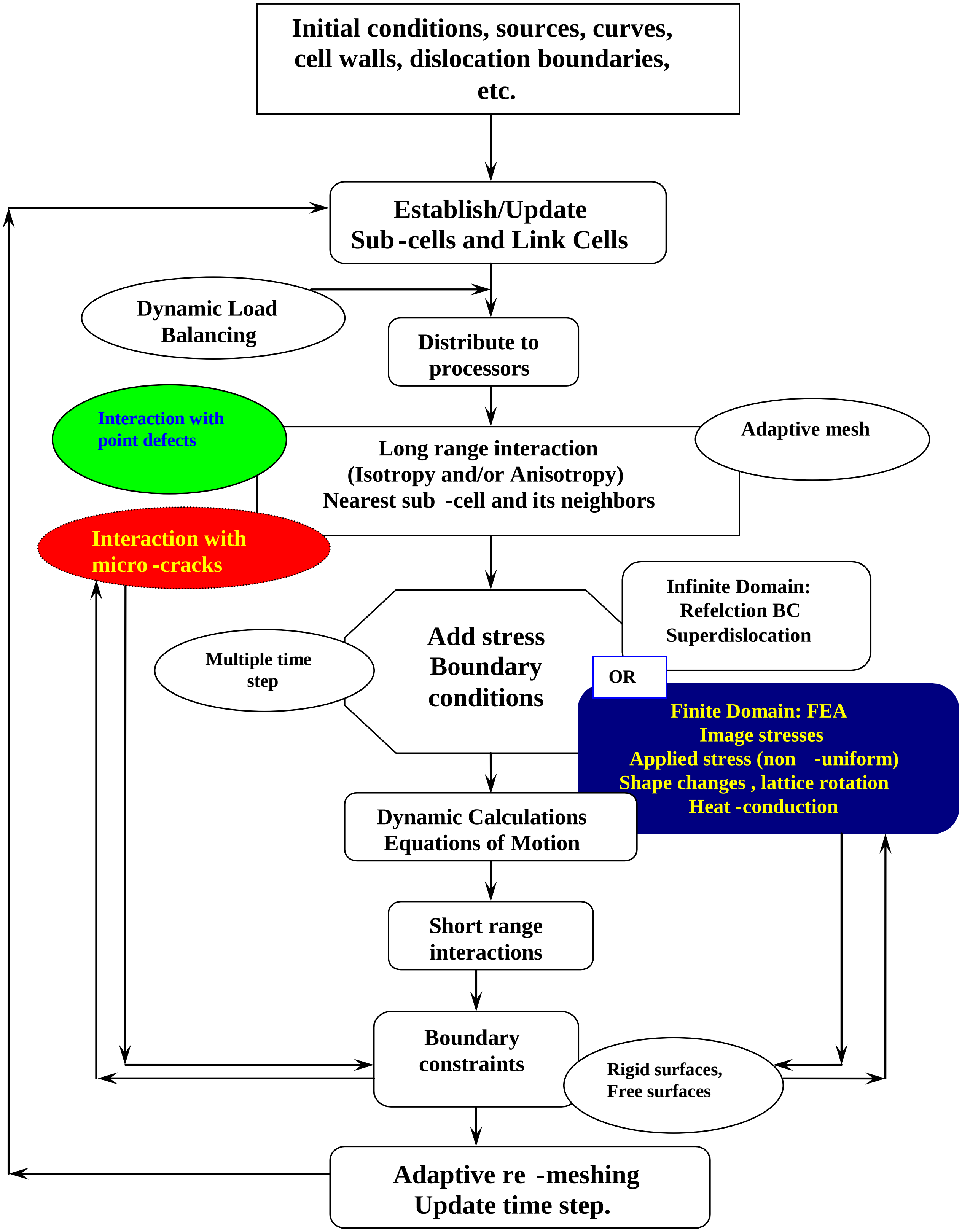
a.3 Control Parameters

Max number of time steps: maxstep (defined in "*data*")

Frequency of re-meshing

Frequency of updating far stress fields and FE analysis (multiple time step)

b. Flow Chart of Program


Initial conditions, sources, curves, cell walls, dislocation boundaries, etc.
Establish/Update Sub -cells and Link Cells
Dynamic Load Balancing
Distribute to processors
Interaction with point defects
Long range interaction (Isotropy and/or Anisotropy) Nearest sub -cell and its neighbors
Adaptive mesh
Interaction with micro -cracks
Infinite Domain: Refelction BC Superdislocation
Multiple time step
Add stress Boundary conditions
OR
Finite Domain: FEA Image stresses Applied stress (non -uniform) Shape changes , lattice rotation Heat -conduction
Dynamic Calculations Equations of Motion
Short range interactions
Boundary constraints
Rigid surfaces, Free surfaces
Adaptive re -meshing Update time step.

c. Execution and Run Options

1. **Dislocations Dynamics ONLY** “*micro3d*”: Infinite domain problems with periodic, or reflected or rigid boundary condition.

   Option in *data:*

   Line 6: 0 0 (DO not execute *fea3d*!!)

2. **Couple finite element analysis with dislocations Dynamics** “*micro3d+fea3d*”

   Finite domain; boundary value problems.
   fea3d can be executed with either static or dynamics options.

   Line 6: 1 1 (1 static or 2 dynamic)

d. *micro3d* Input Data Files Required

micro3d reads from two files:

data
DDinput

e. *fea3d* Input Data Files Required

Option *IndexFE=1(static) or 2(dynamic)*, the following ONE data file must be provided:

FEAconditions

Most of the data is read in the module “initio02.F”. A description of each of these files is presented in following pages.

f. Pre-processing: Input Data Files Generation

A number of Modules are available to generate data for:

a) Random distribution of dislocations and Frank-Read sources in bcc and fcc materials on all slip systems,

b) Random distribution of prismatic loops,

c) Dislocation boundaries (Cell Walls).

g. Output Files: Results and Post-processing

The output data is formatted for use with either ***Techplot*** or ***Gnuplot***

g.1 Techplot Format

Data generated in DDout0.F------>>

sgtecplot.out contains history of dislocation coordinates it can be used to view dislocation motion and to make movies in tecplot.

DDtimeResults.out Contains time dependent results like density, stress, strain etc.; only created if indexFE = 0.

DDsubcellResults.out Contains results for each subcell; only created if indexFE = 0.

Data is generated in *FEA3d*.F---->>

FEAresult.out contains the FEA nodal variables data: mesh coordinates, displacement, stress, strain, and plastic strain tensors.

Other data can be extracted: see also FEAtimedisp.out

h. Specification of Loading Condition Options

h.1 *Micro3d* ONLY

The following lines must be specified:

In *data*

Line 4. *e.g. Specify type of boundary*

0: rigid boundary
1: free or reflection boundary and .
3: Periodic

Line 5. *number of sub-cells*

Line 6. *0 0*

Then two types of loads can be specified:

a. Constant stress (*creep.F*): *The stress is homogenous*

In *input*

Line 2. *specify all, and loadtype=1*

Line 4 *specify stress components*

σ_{xx}, σ_{yy}, σ_{zz}, σ_{yz}, σ_{zx}, σ_{xy}

b. Constant strain rate *(constrain.F)*

In *input*

Line 1. *erate = the strain rate and indexrate = direction of loading (1,2…6)*

Line 2. *specify all, and loadtype=0*

h.2 Couple *micro3d-fea3d*

<u>User subroutines:</u> For use with *fea3d* to specify loading and boundary conditions.

Static analysis (IndexFE=1): Displacement control
Specify Displacement Boundary Condition (**hasan1=1**)at nodes with kfix=2

if(hasan1.eq.1.and.IndexFE.eq.1)then in *fea3d*.F call:

```
subroutine FEAdispStatic(ffext,neeq,kffix,ttt,dttt)
  ! neq     = total number of degrees of freedom
  ! kfix(i) = fixity data
  ! fext(i) = displacement vector
  ! ttt     = current time
  ! dttt    = time increment
 dimension ffext(neeq), kffix(neeq)
   velocity=10e6
   dd=ttt*velocity
  do i=1,neeq
    if(kffix(i).eq.2)then
    ffext(i)=dd
  else
    ffext(i)=0.0
  endif
  enddo
 return
```

Dynamic analysis(IndexFE=2): Velocity control

For velocity of nodes of type kfix=2

if(kfix(i).eq.2) then (called in dynamic.f in *fea3d*.F)

```
subroutine FEAdispDynamic(kf,dt,du)          !dt=time step !kf=node fixity
common /time1/nstep,timenow,dtc,deltt
 velocity = 10e08
 du       = dt*velocity
 return
 end
```

<u>i. Restart File</u>

During execution, dislocation data for *micro3d* is continuously written to:

RESTART.FILE

This file could be used to restart simulation from a previous run.

j. Description of Input Data Files

j.1 Control Data

File: **“*data*”** (read in the routine **initio02.F**)

data

```
1.-crystal------maxstep------sidex,   sidey,     sidez
    'BCC'
               1000000      35000.0   35000.0    35000.0
2.-denisty(kg/m3)---shr(Pa)(MO)--pois--mobility(1/pa.s)--ba(m)
      2700.0       12.3e10      0.309      1.e3        2.725e-10
3.-temper--stkfe(J/m^2)--ismobil--amfactor--thermk(W/m K)-heatc(J/Kg K)
   300.0     0.04     1          0.025      390.       385.
4.-npolorder---ncell-----ifree(0,1,3)----nsface1(3)-----nsface2(3)
     2            0         1             1  1  1       1  1 0
5.-nscx,y,z(nscx.nscy.nscz = number of subcells)
   5  5  5
6.--FiniteElement(IndexFE=0,1,2),indexFE1=0:GenerateData,1:Read, NFEA
    1 0 100
7.—Prismatic-SFT Loops data (0=no defects, 1=Yes),(ndz=1(loops)
                                                 =2(SFT’s), 3=square)
      0  1
8.—Define Coordinate system (w.r.t. crystal axis)
        1.  0.  0.
        0.  1.  0.
        0.  0.  1.
9.--Index for output of results: Every N steps (nndx),
                  gnuformat, tecplotformat (0=NO, 1=Yes)
     50       0       1
10.--IntegOption(IDTdd=0 Const dt,1 Varible), imeshdd(0=cons, 1=auto)
    0 0
```

Block 1: --crystal------maxstep------side

crystal: FCC or BCC (followed by a separate line)
maxstep: maximum number of steps
sidex, y, z: cell size (normalized by the magnitude of the Burgers vector)

Block 2: --density(kg/m^3)---shr(Pa)(MO)--pois--mobility(1/pa.s)--ba(m)

density(rho): material density (kg/m^3)
shr: shear modulus (Pa)
pois: Poisson’s ratio
mobility(amg): dislocation mobility (of edge and mixed) (1/pa.s)
ba(brgmgal) magnitude of burger’s vector (m)

Block 3:-- temper--stkfe--ismobil--amfactor----thermk---heatc

temper: Temperature (K)
stkfe: Stacking Fault Energy (J/m^2)
ismobil: 0 or 1, 0: mobility of edge = mobility of screw,
1: mobility of screw = mobility of edge/mixed * amfactor
amfactor: = (mobility of screw)/(mobility of edge/mixed)
thermk: thermal conductivity (W/m K)
heatc: Specific heat capacity (J/Kg K)

Block 4: .--npolorder----ncell---------ifree-------nsface1(3)------nsface2(3)

npolorder: order of “superdislocation” expansion, =2
ncell: number of reflected cells (0=finite domain)
ifree: 0: rigid boundary, 1: free or reflection boundary, and 3: periodic
nsface1(3) = 1 or 0 (1=Yes free face x, y, z)

nsface2(3) = 1 or 0 (1=yes reflection boundary, x, y, z)

Block 5: .—nscx, y, z (nscx,nscy.nscz=number of subcells for long range stresses)

nscx,y,z 3,4,5..10 (not less than 3!)

Block 6:-- Finite element data

IndexFE = 0 no FEA, =1 Static FEA, =2 Dynamic FEA

IndexFE1 = 0 Generate FE data, = 1 Read FE data

Nfea Number of DD steps per one FEA step

Block 7:—Index for point defect

Loopfile: 0=no loops, 1=read loops

ndz: 1= loops, 2= SFT's, 3=square

Block 8:—Define Coordinate system (w.r.t. crystal axis)

1. 0. 0. (direction of x-axis)
0. 1. 0. (direction of y-axis)
0. 0. 1. (direction of z-axis)
(in this example the cell axes are in the same direction of the crystal axes.)

Block 9: How often the results is printed out and written to **RESTART.FILE** (nndx)
Gnuformat (=1, if gnuplot format output is required),
techplotformat(=1, if techplot format output is required)

50 0 1

Block10.—Integration Option (IDTdd=0 constant time step deltt: =1 variable time step)
Meshing Option (imeshDD=0 constant remeshing,=1 auto remeshing)

---

j.2 Initial Input Data (geometry, connectivity, etc.)

File: **"*DDinput*"** (read in the routine **initio02.F**)

DDinput

Restart: Input could also be **restart data** from a previous run: ***micro3d*** frequently writes a file called "**RESTART.FILE**" containing restart data. For starting from a previous run, rename **RESTART.FILE** to *input* and run ***micro3d***.

```
2002
 1: node       fixed  erate indxerate  jn  jogn  nstep ntotal
      4         10.0  10.0      4      0    0    0     0
 2: timenow  strn stress deltt << dbt  loadtyp 0=constntStrainRate
                                                  1=creep, 2,3..=No DD)
     0.       0.   0.   1.E-11   1.E-07     1
 3: strain increment 6 components
     0. 0. 0. 0. 0. 0.
 4: external stress 6 components  (For Homogenous load, and DD ONLY)
     5.e7   0.   0.   0.   0.   0.
 5: coordinates      and        Burgers vectors, bx, by, bz
  1000.  -5000.  -10000.       -0.5773503   0.5773503   0.5773503
  1000.  -5000.   10000.       -0.5773503   0.5773503   0.5773503
  1000.   5000.   10000.       -0.5773503  -0.5773503   0.5773503
  1000.   5000.  -10000.       -0.5773503  -0.5773503   0.5773503
6: nbr(i,1)     nbr(i,2)  nbr(i,3) PlaneType Constraint ixbtype
     0          2          0     5           7           4
     1          0          0     5           7           4
     0          4          0     6           7           3
     3          0          0     6           7           3
7:Defect Data(#of Defects; followed by Defect size, plane & Coordinate)
  1
  10.  1    2000.   1000.   3500.
```

Line 0: The year the file DDinput was created
Line 1: node fixed erate indxerate jn jogn nstep ntotal

node: the initial total number of nodes
fixed: the initial value of average segment length (in Burgers vector)
erate: strain rate (1/s)
indxerate: strain (or stress) component with strain rate erate
indxerate = 1, 2, 3, 4, 5, 6
ε_{11} ε_{22} ε_{33} ε_{23} ε_{13} ε_{12}
jn: initial number of junction nodes
jogn: initial number of jogs (nodes)
nstep: number of steps already executed (if restarting form an earlier run)
ntotal: number of iterations already executed (if restarting form an earlier run)

Line 2: timenow totalstrn totalstress deltt dbt loadtyp (0 or 1)
timenow:
totalstrn: total strain at nstep
totalstress total stress at nstep
deltt: time step during iteration
dbt: time step (made up the sum of deltt)
loadtyp: =0 for constant strain rate
=1 for constant stress (creep).

Line 3: strain increment 6 components
Line 4: external stress 6 components
Line 5: coordinates and Burgers vector for each node
coordinates: x, y, z for each node
Burgers vecto: x,y,z components for wach node

Line 6: list neighbors of each node, plane type , constraint , Burgers Vector Index (ixbtyp(i))
plane type (iplane(i)): For each node
constraint (icn(i)): For each node (0,1,…)
Burgers Vector index: For each node (1,2,..)

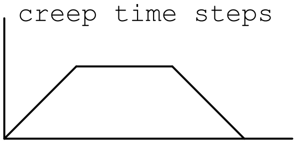
Line 7: Number of defects (if any). followed by Defect size, plane & Coordinates)for each defect.

j.3 Finite Element Data (Special Case! For parallelepiped geometry)

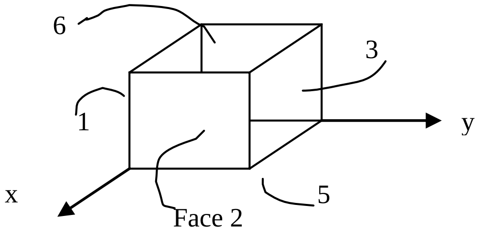
If “IndexFE = 1”, the following FEAinput data file should be provided:

FEAconditions

```
Values for “hasan” and “hasan1” (Type of Displacement BC)
1 0
Values for “Khan” and “Khan1” (Type of Each Boundary)if Hasan=1
0 0 0 0 0 0 Khan(i),  i=1,..6
0 0 0 0 0 0 Khan1(i), i=1,..6
Nall= '0' or 1. 0: Tractions on entire surface. 1: Specify elements
1
DISLOCATIONs image stress "y" or “n”.  “y” or “n” Inanoindentation
y
n
Nall=1: Surface (1-6) & '0' for traction-free or '1' to input traction.
1       0    0.          0.       0
2       0    0.          0.        0
3       1    0.          0.       100000000
4       0    0.          0.       0
5       1    0.          0.       100000000
6       1    0.          0.       100000000
Nanoindentation: Pmn, Pmx, DP, Delta unloading #of creep time steps
0.0  1000e-6 500e-7 500e-7 2
```



```
END of FEAconditions************************
If Nall=0: These lines should Replace line Nall=1 and 6 lines below it
4                      !number of loaded elements
1    3                 !element number…..number of loaded surfaces
   1    16524  0.   0.  !surface number followed by traction
   4    89773  0.   0.
   5   -18974  0.   0.
2    2
   1   90834   0.   0.
   5   87634   0.   0.
3    1
   5   -3847   0.   0.
13    1
   6   0998    0.   0.
```



```
If Hasan=0: These lines should follow after Khan1
7           !No. of constrained nodes.
1 0 0 1
11 0 0 1
16 1 1 1
36 1 0 0
40 1 0 0
63 0 1 0
64 1 1 0
```

```
*****************
*Block line: Hasan, Hasan1
    Hasan=0 1 or 2 ;
         0 >> Line 4: input the total number of the constrained nodes
              followed by node number and its fixity.
         1 >> Line 2: Free/Symmetric/Rigid according to Khan(i),i =1..6
         2 >> Line 3: four corner nodes of the base are fixed
    Hasan1=0 or 1
           0 >> no displacement BC
           1 >> Line3: Non-zero displ. BC according to khan1(i)
                   and user subroutine FEAdispStatic.F(see above)
Block 2:
      Khan(1)..Khan(6)   6 surfaces 0:free, 1:symmetric, 2:rigid
      Khan1(1)..Khan1(6) 6 surfaces 0:free, 1:non-zero NORMAL disp.,
                                    2:non-zero disp. in all DOF

      Note: Both khan(i) and khan1(i) cannot have a non-zero value!
```

9. Appendix A: Data Generation Codes

There are two data generation codes available both for *fcc* and *bcc* materials. These can generate the data file (*input*) for various dislocation structures such as Frank-Read source, planar boundary, dislocation walls, point defects etc.

User Interface: PreMs MDDP02.exe

A-1. Data generation for *fcc* materials: GendataFCC

GendataFCC is a code that generates the initial input data for ***MDDP02*** for ***fcc*** materials, for different arrangements of dislocations, including:

1 = Frank-Read Sources
2 = Array of Frank-Read Sources
5 = Cell walls (Modified Mughrabi Model; need file *walldata*)
6 = Carpets (need file *carpetdata*)
7 = Planar Boundary (need file *planardata*)
10 = Frank-Read Sources decorated with loops
11 = Frank-Sessile loops (circular)

The file ***datain*** is required for all the data types, however 1, 2, 10, and 11 require console input for data generation. 5, 6, and 7 require additional data files. The description of all data files is given in the following section.

Description of Input Data Files

Control Data

File: **"*datain*"**

datain

```
1. Cell Size (Simulation cell size in x-, y-, and z- axes, in units of b)
       5000. 5000. 5000.
2. The coordinate axis coincides with the crystal axis(=0): =1: If you want to
      rotate
       0
3. Enter the directions of the x-axis and y-axis? (z-axis is determined by
     software.)
       1. 0. 0.
       0. 1. 0.
```

Line 1: --Define Cell size in x-, y-, and z- direction (units of Burgers vector)
Line 2: --Define if you want to rotate the coordinate axes
0: don't rotate
1: rotate the axes (to the axes given in Line 3)
Line 3: --Define your coordinate axes (only used if 1 is given in Line 2)
z-axis is determined by the code

Note: The parameters in the file ***data*** (in ***MDDP***) should be consistent with the file ***datain***.

Initial Input Data

File: **"*walldata*"** required for generating data of type 'c'

walldata

```
1. Number of Columns (One Col. is two set of +ve & -ve disls)
       5
2. Number of rows (one column and one row is a dipole)
       3
3. Z-Separation distance between dislocations(1000b)
       1000
4. y-eparation distance between dislocations(500b)
       500
5. Slip plane index (1,2,3, or 4)?
       3
6. Burgers Vector index (1,..6), should coincide with plane!
       5
```

Line 1: --Define number of colums
Line 2: --Define number of rows
Line 3: --Define separation distance in z-axis
Line 4: --Define separation distance in y-axis
Line 5: --Define slip vector (index using Table 4)
Line 6: --Define Burgers vector (index using Table 4)

File: **“*carpetdata*”** required for generating data of type ‘d’

carpetdata

```
1. Number of Columns (One Col. is two set of +&- disls)
       4
2. Number of rows (one column and one row is a dipole'
       3
3. Z-Separation distance between dislocations(1000b)'
       1000
4. y-eparation distance between dislocations(500b)'
       500
5. Slip plane index (1,2,3, or 4)?
       1
6. Burgers Vector index (1,..6), should coincide with plane!
       4
```

Description is same as for ***walldata***.

File: **“*planardata*”** required for generating data of type ‘e’

planardata

```
1. How many sets of dislocations?
       2
2. For this set: How many dislocations?
       10
3. What is the separation distance between two dislocations?
       700
4. What slip Plane (index)
       5
5. What Burgers vector (index)
       2 -1
6. Where do I start (y)
       -2200
7. What is the line direction?
       -0.825 0.351 0.0
2a. For this set: How many dislocations?
       7
3a. What is the separation distance between two dislocations?
       900
4a. What slip Plane (index)
       4
5a. What Burgers vector (index)
```

```
       4  1
6.a Where do I start (y)
       -2400
7a. What is the line direction?
       0.625 -0.105 0.0
```

Line 1: --Define sets of dislocations you want on the boundary
Line 2: --Define number of dislocations lines
Line 3: --Define separation distance between dislocation lines
Line 4: --Define slip plane (index using Table 4)
Line 5: --Define Burgers vector (index using Table 4)
Line 6: --Define starting point for first dislocation (x- and z- starting points are fixed)
Line 7: --Define dislocation line sense
Note: Repeat Line 2-7 for number of dislocation sets you mentioned in Line 1.

Output Files

The output data is formatted for use with either ***Tecplot®*** or ***Gnuplot®***

sg.out — can be used to view dislocations as generated using Gnuplot®.

sgtecplot.out — can be used to view dislocations as generated using Tecplot®.

input — can be used as input to MDDP02.

loopfile.data — this file is created if you are using Point Defects (copy this file to ***MDDP02*** folder along with ***input***, and make sure that line 7 in the file ***data*** is changed accordingly)

A-2. Data generation for *bcc* materials: GendataBCC

GendataBCC is a code that generates the initial input data for ***micro3d*** for ***bcc*** materials, for different arrangements of dislocations, including:

1 = Frank-Read Sources
5 = cell walls (Modified Mughrabi Model; need file walldata)
6 = planar Boundary (need file planardata)
10 = Frank-Read Sources decorated with loops
11 = Frank-Sessile loops (circular)
12 = Lassila’s Case

The file ***datain*** is required for all the data types, however 1, 10, 11, and 12 require console input for data generation. 5 and 6 require additional data files. The description of all data files is the same given in GendataFCC section (Please use Table 3, where Table 4 is mentioned in the description of GendataFCC).

10. Appendix B

B-1 Slip Systems in FCC Metals

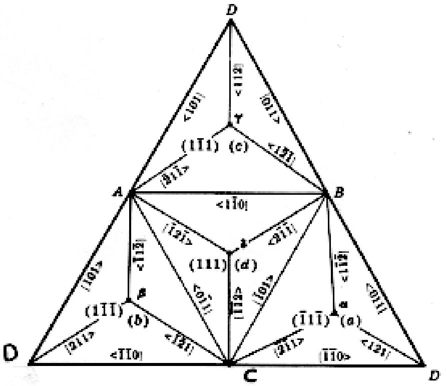


FIGURE 10-10. A Thompson tetrahedron opened up at corner *D*. Both the Thompson notation [(*a*) for glide plane, AB for Burgers vector of perfect dislocation, and Aδ for Burgers vector of partial dislocation] and one possible set of indices for the same planes and directions are presented. The notation (1$\bar{1}$0) is used, instead of the usual notation [1$\bar{1}$0], to indicate the sense of the direction.

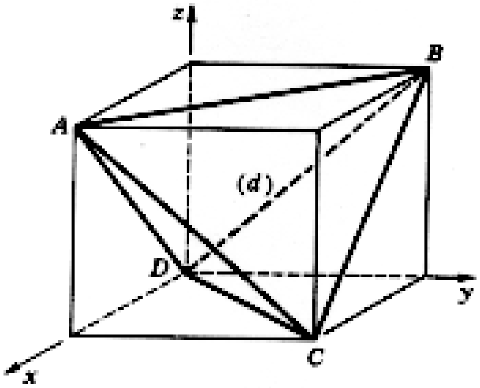


Figure B-1: Slip planes in fcc (Thompson Tetrahedron)

The most closed packed planes for the FCC crystal structure is the {111} family. Slip occurs along <110> type directions within the {111} planes. Table A-1 lists all the possible slip systems in FCC metals.

Table B-1: Slip systems for fcc metals

Planes	(111)	(11$\bar{1}$)	(1$\bar{1}$1)	($\bar{1}$11)
Directions	[1$\bar{1}$0]	[1$\bar{1}$0]	[110]	[110]
	[10$\bar{1}$]	[101]	[10$\bar{1}$]	[0$\bar{1}$1]
	[0$\bar{1}$1]	[0$\bar{1}\bar{1}$]	[0$\bar{1}\bar{1}$]	[$\bar{1}$0$\bar{1}$]

B-2 Slip Systems in BCC Metals

For BCC metals, {110} and {112} are the most closed packed planes. Less closed packed planes are the {123} type family.

Table B-2: Slip systems for bcc metals

Planes	(110)	$(1\bar{1}0)$	(101)	$(10\bar{1})$	$(0\bar{1}1)$	(011)
Directions	$[1\bar{1}1]$	$[111]$	$[\bar{1}11]$	$[111]$	$[111]$	$[1\bar{1}1]$
	$[\bar{1}11]$	$[\bar{1}\bar{1}1]$	$[11\bar{1}]$	$[\bar{1}1\bar{1}]$	$[\bar{1}1\bar{1}]$	$[11\bar{1}]$

Planes	(112)	$(11\bar{2})$	$(1\bar{1}2)$	$(\bar{1}12)$	(121)	$(1\bar{2}1)$
Directions	$[\bar{1}\bar{1}1]$	$[111]$	$[1\bar{1}\bar{1}]$	$[1\bar{1}1]$	$[1\bar{1}1]$	$[111]$
Planes	$(\bar{1}21)$	$(12\bar{1})$	(211)	$(\bar{2}11)$	$(2\bar{1}1)$	$(21\bar{1})$
Directions	$[\bar{1}\bar{1}1]$	$[1\bar{1}\bar{1}]$	$[\bar{1}11]$	$[111]$	$[\bar{1}\bar{1}1]$	$[1\bar{1}\bar{1}]$

Planes	(123)	$(\bar{1}23)$	$(1\bar{2}3)$	$(12\bar{3})$	(132)	$(\bar{1}32)$	$(1\bar{3}2)$	$(13\bar{2})$
Directions	$[11\bar{1}]$	$[\bar{1}1\bar{1}]$	$[\bar{1}11]$	$[111]$	$[1\bar{1}1]$	$[\bar{1}\bar{1}1]$	$[111]$	$[1\bar{1}\bar{1}]$
Planes	(312)	$(\bar{3}12)$	$(3\bar{1}2)$	$(31\bar{2})$	(321)	$(\bar{3}21)$	$(3\bar{2}1)$	$(32\bar{1})$
Directions	$[\bar{1}11]$	$[111]$	$[\bar{1}\bar{1}1]$	$[1\bar{1}1]$	$[\bar{1}11]$	$[111]$	$[\bar{1}\bar{1}1]$	$[1\bar{1}1]$
Planes	(213)	$(\bar{2}13)$	$(2\bar{1}3)$	$(21\bar{3})$	(231)	$(\bar{2}31)$	$(2\bar{3}1)$	$(23\bar{1})$
Directions	$[11\bar{1}]$	$[\bar{1}1\bar{1}]$	$[\bar{1}11]$	$[111]$	$[1\bar{1}1]$	$[\bar{1}\bar{1}1]$	$[111]$	$[1\bar{1}\bar{1}]$

11. Appendix C

Table C-1: Calculated values for the critical stress for the bowout of a single dislocation

Spacing (b)	Segment Length (b)	S (compound) (MPa)	Tilt-Wall (MPa)	Orowan (MPa)
100	30	246.0	203.2	272.0
500	50	65.0	63.9	54.4
900	50	39.0	40.3	30.2
1000	40	36.0		
	50	37.0		
	100	37.0		
	180	37.0	37.0	27.2
2000	100	20.0	22.5	13.6
5000	100	9.5	9.7	5.4
10000	500	5.5		
	1000	5.4	5.7	2.7
15000	500	4.5		
	1000	4.5	4.0	1.8
20000	500	3.5		
	1000	3.5	3.2	1.4

12. Appendix D

Description of Rules and Numerical Implementation

The objective of this appendix is to give a brief description of some of the major issues associated with the computer implementation of basic rules. The code is constructed as a set of subroutines. Each subroutine is modulized so that one needs to make only minor changes, if needed, in order to study for specific applications. The data structure of the main program is discussed along with flow charts associated with major subroutines.

D-1: Non-dimensionalization

The stress components for a curved dislocation are given in closed line integral form. Numerically it is quite expensive to handle the complex nature of the integral. To reduce this complexity we employ a simple approximation method where a curved dislocation is approximated by a set of straight line segments as discussed in Chapter 4. The equations are simple algebraic equations (given in Appendix I) and hence much easier to handle numerically. For the code development, we use a non-dimensionalized approach. For example, the space coordinates are normalized by the magnitude of the Burgers vector b with each unit distance being the magnitude of the Burgers Vector as χ = r/b, where r is the vector from dislocation segment to a point of interest. For example, the stress field is inversely proportional to r as

$$\sigma \propto \frac{b}{r}\mu \tag{D-1}$$

where μ is the shear modulus. With $\hat{x}$ = r/b, equation (C-1) becomes

$$\overline{\sigma} = \frac{\mu}{\hat{r}} \tag{D-2}$$

Similarly, the net driving force per unit length

$$\mathbf{F} \propto \sigma b \Rightarrow \hat{F} \equiv \frac{F}{b} = \sigma \tag{D-3}$$

with a unit of force per unit length per Burgers vector. Also, let the non-dimensional velocity $\hat{v}$ = v/b then we have

$$\hat{v} = B\left(\frac{F}{b}\right) \Rightarrow \hat{v} = B\hat{F} \tag{D-4}$$

with the time increment of Δt we obtain the glide distance as

$$\hat{x} = \hat{v}\Delta t \tag{D-5}$$

using equations (C-1)-(C-4), although it is a small gain in the computational aspect, there is no need for the magnitude of b. For the actual values of velocity and net force, b should be multiplied.

D-2: Data Structure

Identification of Basic Geometry: To identify a dislocation segment in 3D space, we need the following:

a) coordinates of two end points of a dislocation segment
b) neighboring nodes
c) line sense vector
d) Burgers vector
e) plane index
f) cross-slip index
g) constraints

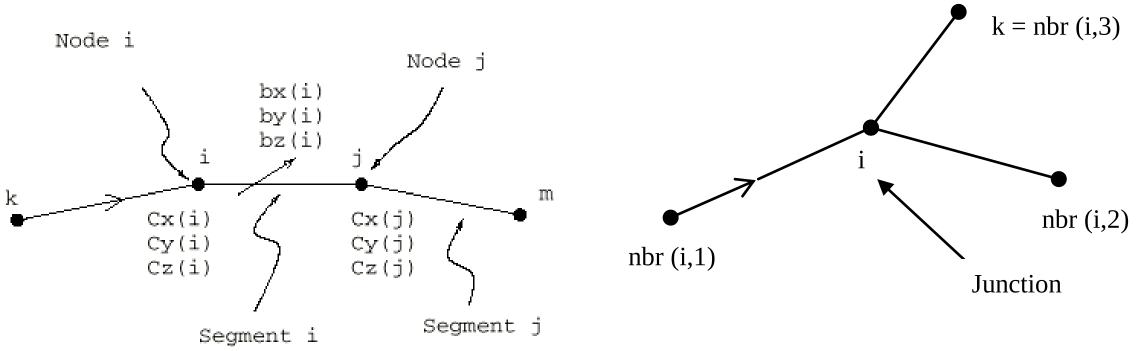


Figure D-1: Dislocation segments and nodes

For example, consider a set of dislocation segments as shown in Figure C-1. In the figure, the segment number is represented by the node from which dislocation line begins; i.e. segment i is represented by node i. The line sense vector is determined using the neighboring nodes. For each node number it has two neighboring node numbers, i.e.,

nbr(i,1), nbr(i,2),

where i is the node number and index 1 and 2 represents the backward and forward node, respectively. The forwarding neighbor of segment i (in the direction of the line sense) in the figure is j, i.e.

$$nbr(i,2) = j$$

and the backward neighbor of segment i is m as

$$nbr(i,1) = k.$$

To specify a junction node as shown in the figure, we use its third neighbor node as nbr(i,3). Initially, the value of nbr(i,3) is set zero. When a junction is formed the value of nbr(i,3) becomes the node number of the segment whose end node is attached to the junction node. For example, the junction node of node i is nbr(i,3) = k as shown in the figure. The line sense vector is simply obtained by considering the two end nodes for a given segment. The plane index is introduced to specify the type of plane in which the dislocation segment can glide. The plane index for different slip planes is given in Chapter 6. Due to the geometrical constraint, we specify a number of different constraint types since there are many constraints associated with the behavior of dislocation motion. Table 2 lists the type of each constraints and corresponding index values of an array icn(inode).

For free boundary conditions, segments reaching boundaries must disappear. This introduces another block of statements that must account for the node elimination on the crystal surface. Nodes on the surface are constrained to move along its crystallographic directions. In bcc metals, for example, since the unit vectors specifying the six (110) and twelve (112) planes are stored in the beginning of the program, one can simply use this information to treat the motion of nodes on the surfaces of the crystal. For example, Figure C-2 shows nodes on the boundary along its crystallographic directions. In the figure the dislocation line is in (110) plane which has the plane index of 5. There are two end nodes on the (010) and (001) surface, xy and xz planes, respectively. The node on the xy plane is constrained to move along the line x = y, which has the constraint type 3. i.e. icn(inode)=3. If a node on the (110) plane has the constraint type 2, it is constrained to move in the z direction on the xz plane as shown in the figure. For rigid boundary condition, dislocation segments pile-up at the boundary.

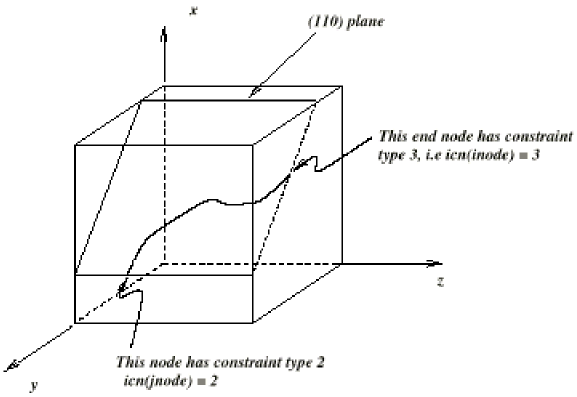
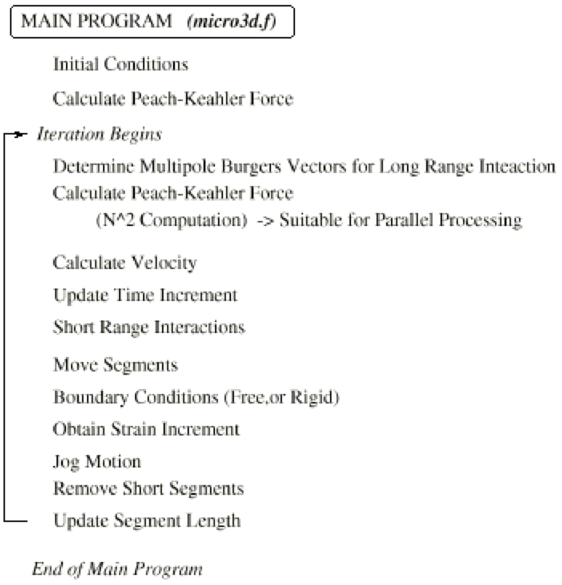

x
(110) plane
This end node has constraint type 3, i.e icn(inode) = 3
z
y
This node has constraint type 2 icn(jnode) = 2

Figure D-2: Boundary nodes

D-3: Main Features

The program is called *micro3d* which consists of one main program and a number of subroutines as shown in Figure C-3. First, the main program reads the material property parameters from the *input* file for the initial condition such as the type of crystal (bcc or fcc), material constants, cross slip data structures. Then it determines, based on the flag value, whether to perform constant strain rate test or creep test. Once all information is gathered, it calls the subroutine that calculates the Peach-Koehler force, including interaction from remote segments, adjacent segment force calculation including line tension, and long range interaction using superdislocations. The velocities are based on different values for the mobilities of screw and edge character, giving an option for the same and different mobility. The net force acting on each segment is stored in a vector array. The values of velocities are also stored for a later usage when nodes are moved according to the product of average amount of velocities of two segments and the time increment. Then, the features of short-range interactions are checked followed by the calculation of the plastic strain increment and movement of nodes. This is summarized in a flow chart in Figure C-4.



MAIN PROGRAM *(micro3d.f)*

Initial Conditions

Calculate Peach-Keahler Force

Iteration Begins

Determine Multipole Burgers Vectors for Long Range Inteaction

Calculate Peach-Keahler Force

(N^2 Computation) -> Suitable for Parallel Processing

Calculate Velocity

Update Time Increment

Short Range Interactions

Move Segments

Boundary Conditions (Free,or Rigid)

Obtain Strain Increment

Jog Motion

Remove Short Segments

Update Segment Length

End of Main Program

Figure D-3: *micro3d.f* code control

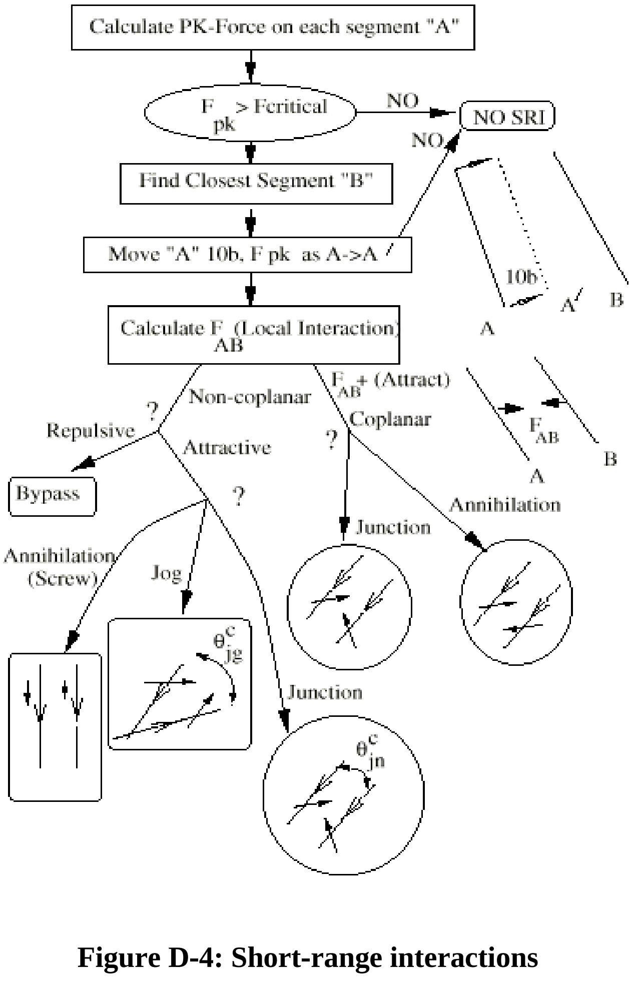


Figure D-4: Short-range interactions

D-4: Cross-slip

Cross-slip is an important mechanism in recovery processes in both fcc and bcc metals. Screw dislocations may cross slip to reduce internal stresses and to circumvent internal obstacles, consequently, providing a mechanism for the production of Frank-Read sources through double cross-slip. The process is very prolific in bcc materials due to the availability of many secondary slip systems; a <111> screw dislocation could cross-slip on three {110} planes, three {112} planes and six {123} planes. The ease of a screw dislocation to cross-slip has been observed in stage I in the form of "composite slip" or wavy slip lines (Mitchell and Spitzig, 1975). This mechanism was first proposed by Taylor and Elam (1926) and was referred to as "pencil glide" to explain the wavy slip traces in iron. They suggested that although the slip plane was not clearly defined the slip direction was clearly <111>. Subsequent investigations have suggested that the observed wavy glide is the result of cooperative cross-slip occurring in increments of a few to a thousand Burgers vector units on appropriate {110} and {112} planes.

For the <111>{110} and <111>{112} slip systems in bcc, e.g., a <111> screw dislocation is common to six different planes. Within the present framework of discrete dislocation segments, if a screw dislocation segment cross-slip to a secondary plane it would have to bow-out to form a "Super-kink" configuration as shown in Figure C-5.

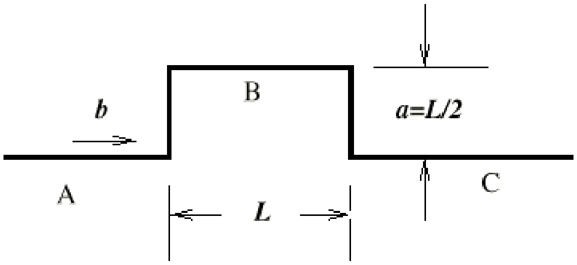


Figure D-5: Model for cross-slip mechanism

Thus, for cross-slip to take place the segment would have to overcome a barrier whose strength is determined by the elastic activation energy for that configuration. However, cross-slip is a thermally activated process and is determined numerically using a Monte-Carlo type simulation as explained below. The probability of a segment to jump into a secondary plane is determined by the probability *P* as

$$P = \alpha\, \Omega_1\, \delta t exp\left(-\frac{\Delta W^* - \tau A}{kT} \right) = \Omega_1\, \delta t exp\left(-\frac{(\tau^* - \tau)A}{kT} \right), \quad \Omega_1 = \frac{C_t \pi}{L} \qquad \text{(D-6)}$$

where Ω_1 is the fundamental frequency of a vibrating dislocation segment of length *L*, C_t is the transverse sound velocity, δt is the time increment, α is a numerical parameter controlling the frequency of cross slip, ΔW^* is the double kink activation energy, τ is the resolved shear stress, *A* is the area swept by the dislocation segment, *k* is the Boltzmann constant , and T is the absolute temperature. The activation energy based on the double kink as shown in Figure C-5 is given in [14], and τ^* is the corresponding critical stress to form the critical configuration. This configuration (with *a=L/2*) corresponds to the (approximate) critical configuration to bow out a pinned dislocation to a semi-circle configuration. With this condition, once the dislocation segment is moved into this configuration it will continue to bow out. Otherwise if the segment is moved a small fraction of *L/2* it may retreat due to a large back force from line tension. Moreover, it is argued that since the process is thermally activated, once the barrier is overcome the dislocation would jump to this configuration over a time scale much smaller than that of the simulation time scale.

The for a given segment length, τ^* is found by minimizing the total free energy ΔG of the bow-out shown in Figure 3, where $\Delta G = \Delta W - \tau b A$, $A = L^2 / 2$ being the area swept by the dislocation segment as it forms the kink shown in Figure C-5, for which the ΔW is given by (Hirth and Lothe, 1982, p. 243).

$$\begin{aligned}\Delta W = {} & \frac{\mu b^2}{2\pi}\left[\sqrt{L^2 + a^2} - L - a + L\,\ell n\left(\frac{2L}{L + \sqrt{L^2 + a^2}}\right)\right] \\ & - \frac{\mu b^2}{4\pi(1-\nu)}\left[2L - 2\sqrt{L^2 + a^2} + 2a\,\ell n\left(\frac{a + \sqrt{L^2 + a^2}}{L}\right)\right] \\ & + \frac{\mu b^2 a}{2\pi(1-\nu)}\ell n\left(\frac{a}{e\rho}\right)\end{aligned} \tag{D-7}$$

Upon minimizing ΔG with respect to *L*, we obtain the critical stress τ^* for a given segment length *L*. For Ta, the result is given in Figure C-6

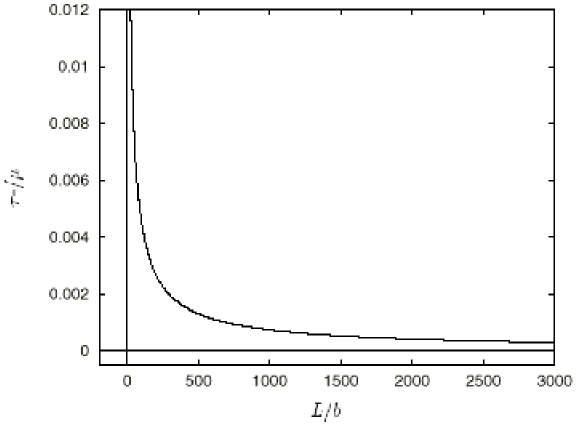


Figure D-6: Critical stress versus segment length for cross-slip.

The Monte-Carlo simulation to determine cross--slip is developed as follows. At a given stress state the shear stress τ is determined for each plane and the probability is evaluated using equation (10) for each of the planes. Then a random number between 0 to 1 is selected to determine which plane the probability falls into. An actual simulation of cross-slip and composite slip is given in Figure C-7. In this simulation, a Frank-Read source is situated on the (011) plane with $[1\bar{1}1]$ Burgers vector as shown in the figure. Then an axial stress σ is applied in the [010] direction. This orientation results into the same resolved shear stress for both (011) and (110) planes. Initially, the dislocation line is oriented such that the dislocation is pure edge. The dislocation bows out around the source and dislocation segments on the far right and far left of the loop become pure screws. In this orientation, the glide force acting on the screw dislocation on each of the possible six glide planes (110), (011), $(10\bar{1})$, $(\bar{1}12)$, $(21\bar{1})$ and (121) is

$\sigma b/\sqrt{6}$, $\sigma b/\sqrt{6}$, 0, $\sigma b\sqrt{5}/6$,$\sigma b\sqrt{5}/6$, and $\sigma b/3\sqrt{2}$, respectively. Therefore, the largest glide force is in the (110) and (011) planes. Moreover, the planes ($\bar{1}12$) and ($21\bar{1}$) experience 86% of this largest glide force. Therefore, the screw dislocation under consideration would have a high probability to cross on any of these four planes. For a relatively high applied stress (200 MPa) cross slip takes place as can be seen in Figure C-7. In this example, the mobility of screw dislocation was assumed be equal to that of the edge dislocation for high temperatures [$M_{ge} = M_{gs} = 10^4$ (Pa.s)$^{-1}$].

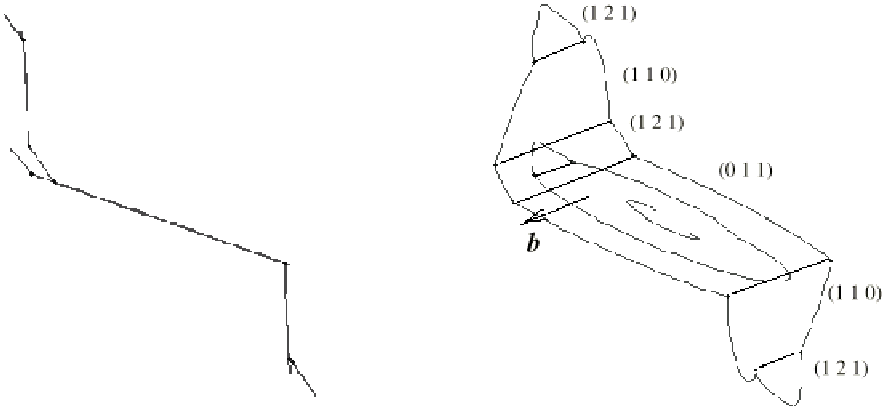


Figure D-7: Simulation of cross-slip

D-5: Annihilation:

Two attractive parallel dislocations in the same plane can annihilate if the force criterion mentioned in Chapter 3 is satisfied. For the actual implementation the following conditions are checked. First, b_1 must be parallel to b_2, i.e. $b_1 \times b_2 = 0$. Second, the line sense vector ξ_1 and ξ_2 have to be force parallel ($\xi_1 \times \xi_2 = 0$). Once these two conditions are met, the quantity of net force between two segments is evaluated to determine whether the interaction between them is attractive or repulsive. Consider two attractive dislocation segments as shown in Figure C-8(a). For annihilation, node i should be the forwarding node of node m. Also, node j becomes the forward node on node k, where now k becomes the backward node of j node as shown in Figure C-8 (b). In the program this is implemented as

nbr(i,2) = m

nbr(m,1) = i

nbr(j,1) = k

nbr(k,2) = j

Once annihilation has occurred as shown in the figures, the interaction force between the two segments joined by the sharp corner is very high it is a nearly instantaneous process they become attracted and annihilated. For this purpose, we simply check the angle between the two segments as shown in Figure C-9. If it is smaller than a critical angle we remove the node and rearrange the arrays as

nbr(i,2) = k

nbr(k,1) = i

call remove(j),

where subroutine remove is used to re-shuffle the memory space of empty array addresses, where we move the content in the last node into the node just removed since the total number of nodes after the step mentioned above becomes on node less.

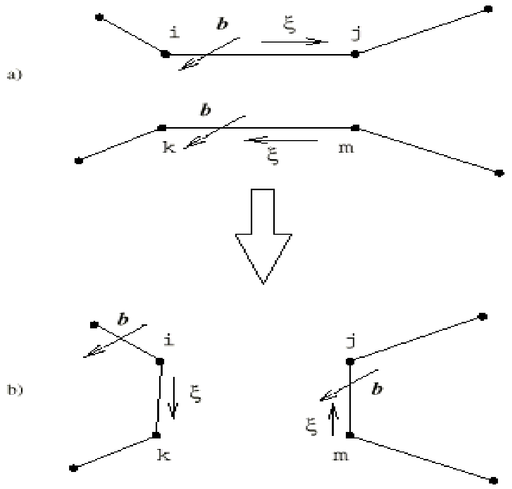


Figure D-8: Annihilations of two segments

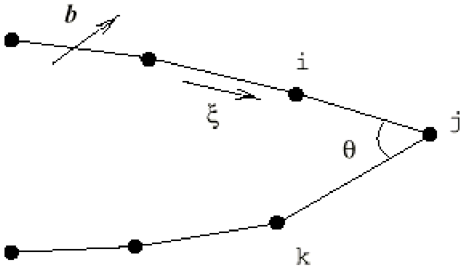


Figure D-9: Reducing sharp corners

Another case that needs to be dealt with is the case where two long dislocations annihilated producing a configuration as shown in Figure C-10(a)-(b). The loop created by annihilation of two attractive segments gets annihilated naturally with a few subsequent iterations.

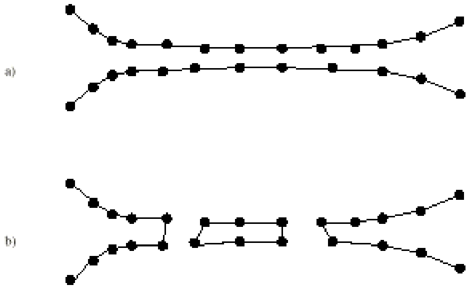


Figure D-10: Annihilation of loops

D-6: Jog

Jogs are formed when the angle between two attractive dislocations in different planes becomes less than a critical angle as discussed in Section 5.5 can form jogs. When the two dislocations are repulsive but one is highly mobile due to other external sources, jogs can be also formed. Figure C-11 (a)-(b) illustrates jog formation of two attractive dislocations. In Figure C-11(a), segment i and segment m are to react to form jogs. When jogs are formed as shown in Figure C-11(b), segment i (nodes i and j) is placed on the other side of segment m (nodes m and n) making the constraint type of node i a jog constraint type, i.e.

icn(i) = 9

Segment m is also placed over the segment i with icn(m) = 9. Due to the length scale of our discretization approach, the actual jog creation is not very desirable, although possible. The jog nodes are treated in a special way by their constraint type. They become immobile until the two adjacent segments reach a critical value. Jogs approaching the crystal surface are annihilated at the boundaries if free boundary conditions are used. At every iteration, we need to check whether if this condition is met. We can simply scan, rather than checking all nodes, only up to the number of jogs in stead of checking all the segments with the constraint type of jogs, i.e. icn(i) = 9 by introducing arrays as

jog(1) = i   jogpoint(i) = 1
jog(2) = m   jogpoint(m) = 2
⋮

The index of jog vector above is the number of jogs. For jog(2) = m, it implies that the second jog index (2) has node number m. The vector jogpoint contains the value of jog indices. For example, the expression jogpoint(i) = 1 indicates that node i has the jog number 1.

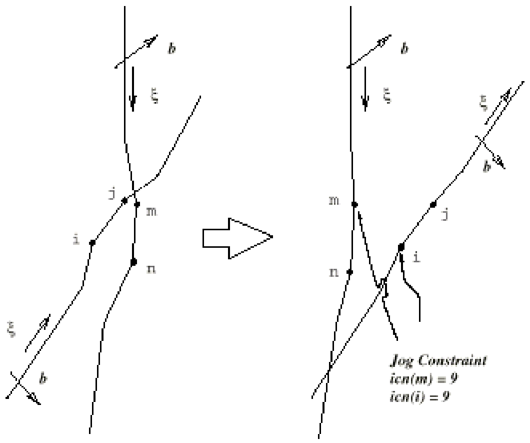


Figure D-11: Formation of jogs

Jogs can move by creating vacancies or interstitials. This strength is balanced by the total line tension of two adjacent segments around the jog node. When the line tension force exceeds the jog strength, the jog moves by the following scheme. (See Rhee et al., (1998) for detailed analysis for this critical jog-motion-angle criterion). When the angle reaches a critical value the jog node is advanced according to the average velocity of two adjacent segments.

D-7: Junction

For the junction formation mechanism, consider two dislocation segments shown in Figure C-12 (a). When Rule 3 in Section 5.4 is satisfied, a junction is formed with the Burgers vector being the sum of those of the two segments as shown in Figure C-12(b). Before junction reaction, node i has its forward neighboring node j with the Burgers vector b_1. Node m has the forward neighboring node n as shown in Figure C-12(a). When a junction is formed the vectors of the neighbors become

```
nbr(i,2) = nbr(n,2)
nbr(i,3) = j
nbr(nbr(n,2),1) = i
nbr(j,1) = nbr(m,1)
nbr(nbr(m,1),2) = j
nbr(j,3) = i
call remove (m)
call remove (n)
```

For non-coplanar junction, one must find the line of two intersecting planes since junction can only form along the common lone of intersectionJunction nodes formed by dislocations in different planes move, by further reactions between the two adjacent segments around a junction node when energetically favorable, only in the direction of the common line of two intersecting planes.

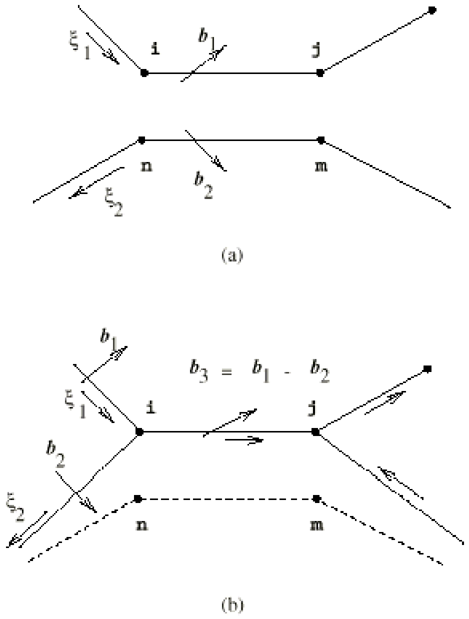


Figure D-12: Junction formation

Two new vectors are introduced to deal with interaction between the junction segment and remote segments. For example, there are example junction nodes created in the figure. These arrays are necessary to avoid any redundant steps for the interaction calculation, i.e. rather than going through the entire segments, only up to the total number of junction nodes can be checked. The two arrays are

jnpoint(1) = i	jnindex(i) = 1
jnpoint(2) = j	jnindex(j) = 2

The interaction forcer on the three segments around a junction node is treated using above vector array.

D-8: Strain Calculation

For the incremental plastic strain calculation, the old position of all segments need to be stored because information of how much each dislocation has moved should be available to calculate the strain increment. The calculation is based on the relation

$$\Delta\varepsilon = \sum_{i=1}^{N} \frac{A_i}{2V}\left(n_i \otimes b_i + b_i \otimes n_i\right). \tag{D-10}$$

where A_i is the area swept by a dislocation segment, n_i is the unit vector normal to the slip plane, b_i is the Burgers vector and V is the cell volume. Similarly, the rotation tensor is given by

$$\Delta\omega = \sum_{i=1}^{N} \frac{A_i}{2V}\left(n_i \otimes b_i - b_i \otimes n_i\right). \tag{D-11}$$

For more realistic simulations, this feature should be included to account for rotation of slip planes towards the loading axis. In our simulation the rotation effects are not included.

D-9:Parallel Processing

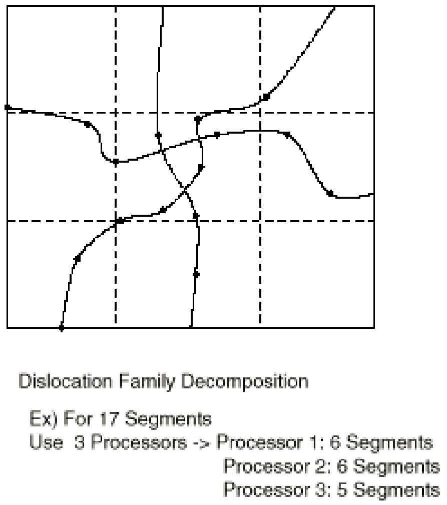
Parallel processing requires an interface daemon across workstations to transport job control and data messages, and for process management. There are two major packages for parallel processing that have been developed and standardized depending on its optimality, portability and communication time. One is the PVM (Parallel Virtual Machine), where its use is intended for a network of heterogeneous workstations. The other is the MPI (Message Passing Interface). The intention of the development of MPE is to provide a standard message passing specification for a specific MPP (Massively Parallel Processor) machines. One major difference between PVM and MPI is that MPI does not include features such as job control and machine configuration etc., but it provides a complete set of functions for message passing. A parallel version of *micro3d* is available using both interfaces. For a heterogeneous network of system architectures, PVM is more favorable due to its portability for different computer architecture. For massively parallel machines, MPI is more commonly used mainly due to its high capability of communication time between processors.

Domain Decomposition Method

Two major methods are commonly used for parallel computing. One is the domain decomposition method, where each processor is responsible for the interaction calculation of dislocations in each sub-space. One disadvantage of this method is that dislocations may form localized dense regions, causing one processor to be responsible for more dislocation segments is its assigned domain than others that have less number of dislocations in their domain. To minimize the waiting time of CPU's, methods which would yield a better computational efficiency, such as dynamic load balancing, adaptive meshing and link cell method may be required for more efficient distribution of computation tasks to all processors.

Dislocation Family Decomposition Method

The other method is the dislocation family decomposition method. In this method equal number of dislocation segments are sent to each processor as illustrated in Figure C-13. This method is rather simple that it can be implemented without much effort. The parallel code is implemented using the MPI software on the IBM RS/6000 system, which has 168 nodes, of which each node has 4 CPU's with the clock speed of 333 MHz. In principle, we can distribute the work load for every do-loop in the program if no data dependence exists inside the loop (although we can re-order the data dependent do-loops for parallel computation, but this exercise has not been extensively studied). The approach we take is based on the simple master-slave scheme, where the master processor sends tasks to slave processors and gathers information from slave processors upon finishing work. A parallel flow chart for our code is given in Figure C-14. The major contribution of CPU usage in the figure is from the long-range interaction calculations, yielding an order of $\boldsymbol{N}^2$ computations. To illustrate how much time is saved using parallel processing, the calculation of the bowout example shown in Figure C-15 was performed with about 2200 segments. At each time increment, each processor was assigned the task of calculating interactions, driving forces, velocities and new configuration for an equal fraction of the total segments. This information is thin relayed to the master machine. The calculation is continued until several loops are generated. The speed-up versus the number of processor is given in Figure C-15. The gain in the CPU time slows down as the number of processor used reaches 100. This is due to the combination of the reduction in the work load of each processor and more frequent data communication between the master and slave processors. As the number of dislocation segments increases this saturation will occur at higher number of processors. However, it can be deduced from the figure that significant amount in CPU time can be achieved.


Dislocation Family Decomposition
Ex) For 17 Segments
Use 3 Processors -> Processor 1: 6 Segments
Processor 2: 6 Segments
Processor 3: 5 Segments

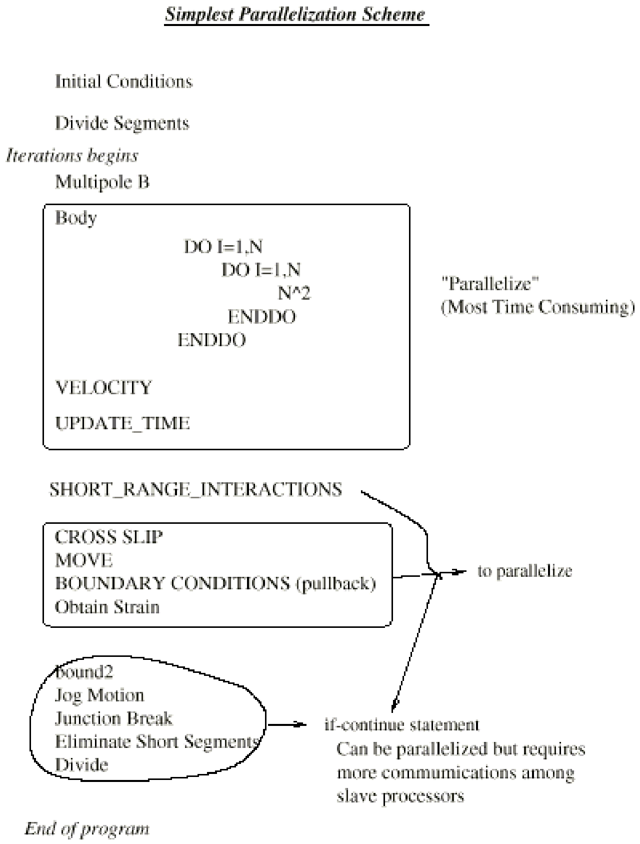


Figure D-13: Family decomposition

Figure D-14: Parallelization scheme

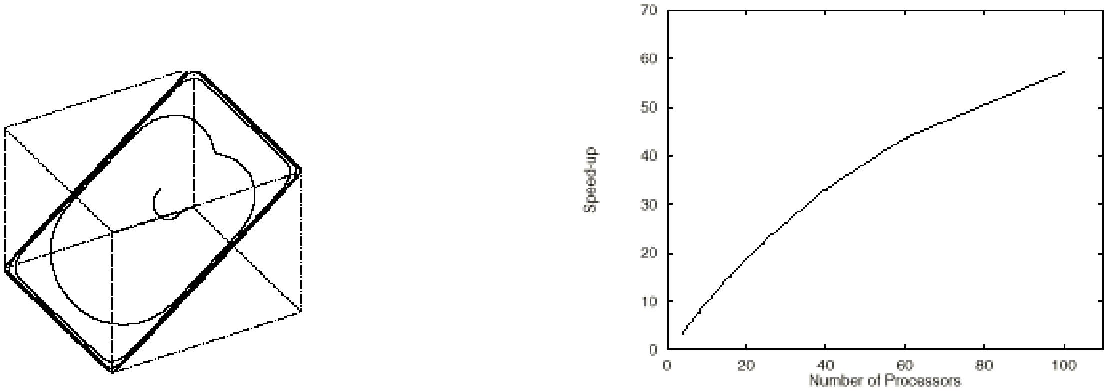


Figure D-15: Speed-up versus number of processors

13. Appendix E

Two equivalent formulae are implemented in the code, the Hirth and Lothe formulae and the de Wit formulae. The de Wit Formulation is more numerically more efficient since the equations are expresses in terms a global reference frame and, therefore, does not require matrix transformation.

E-1. Stress Field About a Finite Segment (Hirth and Lothe Formulae)

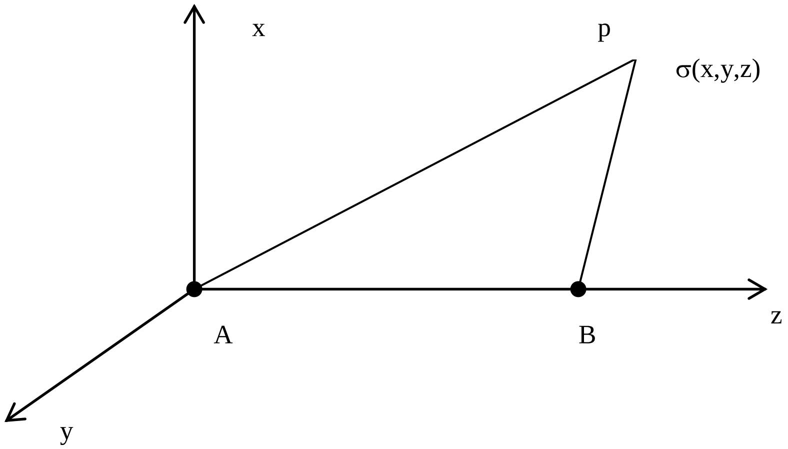


Figure E-1: Stress field around a segment

The stress field $\sigma_{ij}^{P}(A)$ at point P due to the segment AB is given by

$$\sigma_{ij}^{P} = \sigma_{ij}(B) - \sigma_{ij}(A) \qquad \text{(E-1)}$$

where $\sigma_{ij}(A)$ or $\sigma_{ij}(B)$ has its components as listed below.

Set I:

$$\frac{\sigma_{xx}}{\sigma_o} = b_x \frac{y}{R(R+\lambda)}\left(1+\frac{x^2}{R^2}+\frac{x^2}{R(R+\lambda)}\right) + b_y \frac{x}{R(R+\lambda)}\left(1-\frac{x^2}{R^2}-\frac{x^2}{R(R+\lambda)}\right)$$

$$\frac{\sigma_{yy}}{\sigma_o} = -bx \frac{y}{R(R+\lambda)}\left(1-\frac{y^2}{R^2}-\frac{y^2}{R(R+\lambda)}\right) - by \frac{x}{R(R+\lambda)}\left(1+\frac{y^2}{R^2}+\frac{y^2}{R(R+\lambda)}\right)$$

$$\frac{\sigma_{zz}}{\sigma_o} = b_x\left(\frac{2vy}{R(R+\lambda)}+\frac{y\lambda}{R^3}\right) + b_y\left(-\frac{2vx}{R(R+\lambda)}-\frac{x\lambda}{R^3}\right)$$

$$\frac{\sigma_{xy}}{\sigma_o} = -b_x \frac{x}{R(R+\lambda)}\left(1-\frac{y^2}{R^2}-\frac{y^2}{R(R+\lambda)}\right) + b_y \frac{y}{R(R+\lambda)}\left(1+\frac{x^2}{R^2}+\frac{x^2}{R(R+\lambda)}\right)$$

$$\frac{\sigma_{xz}}{\sigma_o} = -b_x \frac{xy}{R^3} + b_y\left(-\frac{v}{R} + \frac{x^2}{R^3}\right) + b_z \frac{y(1-v)}{R(R+\lambda)}$$

$$\frac{\sigma_{yz}}{\sigma_o} = b_x\left(\frac{v}{R} - \frac{y^2}{R^3}\right) + b_y \frac{xy}{R^3} - b_z \frac{x(1-v)}{R(R+\lambda)} \tag{E-2}$$

where $\sigma_o = \dfrac{\mu}{4\pi(1-v)}$, $\lambda = z'-z$ and $R^2 = x^2 + y^2 + z^2$

Two other equivalent forms are:

Set II:

$$\frac{\sigma_{xx}}{\sigma_o} = -b_x \frac{y}{R(R-\lambda)}\left(1 + \frac{x^2}{R^2} + \frac{x^2}{R(R-\lambda)}\right) - b_y \frac{x}{R(R-\lambda)}\left(1 - \frac{x^2}{R^2} - \frac{x^2}{R(R-\lambda)}\right)$$

$$\frac{\sigma_{yy}}{\sigma_o} = bx \frac{y}{R(R-\lambda)}\left(1 - \frac{y^2}{R^2} - \frac{y^2}{R(R-\lambda)}\right) + by \frac{x}{R(R-\lambda)}\left(1 + \frac{y^2}{R^2} + \frac{y^2}{R(R-\lambda)}\right)$$

$$\frac{\sigma_{zz}}{\sigma_o} = b_x\left(\frac{-2vy}{R(R-\lambda)} + \frac{y\lambda}{R^3}\right) + b_y\left(\frac{2vx}{R(R-\lambda)} - \frac{x\lambda}{R^3}\right)$$

$$\frac{\sigma_{xy}}{\sigma_o} = b_x \frac{x}{R(R-\lambda)}\left(1 - \frac{y^2}{R^2} - \frac{y^2}{R(R-\lambda)}\right) - b_y \frac{y}{R(R-\lambda)}\left(1 + \frac{x^2}{R^2} + \frac{x^2}{R(R-\lambda)}\right)$$

$$\frac{\sigma_{xz}}{\sigma_o} = -b_x \frac{xy}{R^3} + b_y\left(-\frac{v}{R} + \frac{x^2}{R^3}\right) - b_z \frac{y(1-v)}{R(R-\lambda)}$$

$$\frac{\sigma_{yz}}{\sigma_o} = b_x\left(\frac{v}{R} - \frac{y^2}{R^3}\right) + b_y \frac{xy}{R^3} + b_z \frac{x(1-v)}{R(R-\lambda)} \tag{E-3}$$

In Eq. (D-3), the $\dfrac{1}{R(R+\lambda)}$ term can be written as

$$\frac{1}{R(R+\lambda)} = \frac{R-\lambda}{R(R^2-\lambda^2)} = \frac{R-\lambda}{Rp^2} = \frac{1}{p^2} - \frac{\lambda}{Rp^2} \tag{E-4}$$

Upon substituting, the first part of σ_{xx} term, for example, can be written as

$$b_x \frac{y}{R(R+\lambda)}\left(1 + \frac{x^2}{R^2} + \frac{x^2}{R(R+\lambda)}\right)$$

$$= b_x y\left(\left(\frac{1}{p^2} - \frac{\lambda}{Rp^2}\right) + \frac{x^2}{R^2}\left(\frac{1}{p^2} - \frac{\lambda}{Rp^2}\right) + x^2\left(\frac{1}{p^2} - \frac{\lambda}{Rp^2}\right)\left(\frac{1}{p^2} - \frac{\lambda}{Rp^2}\right)\right)$$

$$= b_x y\left(-\frac{\lambda}{Rp^2} + \frac{x^2}{R^2p^2} - \frac{\lambda x^2}{R^3p^2} - \frac{2x^2\lambda}{Rp^4} + \frac{x^2\lambda^2}{R^2p^4} + \frac{1}{p^2} + \frac{x^2}{p^4}\right)$$

$$= b_x y\left(-\frac{\lambda}{Rp^2} - \frac{\lambda x^2}{R^3 p^2} - \frac{2x^2\lambda}{Rp^4} + \frac{x^2}{p^4} + \frac{1}{p^2} + \frac{x^2}{p^4}\right)$$

$$= -\frac{b_x y\lambda}{p^2 R}\left(1 + \frac{x^2}{R^2} + \frac{2x^2}{p^2}\right) + b_x y\left(\frac{2x^2}{p^4} + \frac{1}{p^2}\right) \qquad \text{(E-5)}$$

The constant term $b_x y\left(\frac{2x^2}{p^4} + \frac{1}{p^2}\right)$ drops when Eq. (D.1) is used. Therefore we can write Set III:

$$\frac{\sigma_{xx}}{\sigma_o} = -b_x\,\frac{y\lambda}{p^2 R}\left(1 + \frac{x^2}{R^2} + \frac{2x^{\,2}}{p^2}\right) - by\,\frac{x\lambda}{p^2 R}\left(1 - \frac{x^2}{R^2} + \frac{2x^2}{p^2}\right)$$

$$\frac{\sigma_{yy}}{\sigma_o} = b_x\,\frac{y\lambda}{p^2 R}\left(1 - \frac{y^2}{R^2} - \frac{2y^2}{p^2}\right) + by\,\frac{x\lambda}{p^2 R}\left(1 + \frac{y^2}{R^2} + \frac{2y^2}{P^2}\right)$$

$$\frac{\sigma_{zz}}{\sigma_o} = b_x\left(-\frac{2vy}{Rp^2} + \frac{y\lambda}{R^3}\right) + b_y\left(-\frac{2vx}{Rp^2} - \frac{x\lambda}{R^3}\right)$$

$$\frac{\sigma_{xy}}{\sigma_o} = b_x\,\frac{x\lambda}{p^2 R}\left(1 - \frac{x^2}{R^2} - \frac{2x^2}{p^2}\right) - by\,\frac{y\lambda}{p^2 R}\left(1 - \frac{x^2}{R^2} + \frac{2x^2}{p^2}\right)$$

$$\frac{\sigma_{xz}}{\sigma_o} = -b_x\,\frac{xy}{R^3} + b_y\left(-\frac{v}{R} + \frac{x^2}{R^3}\right) + b_z\,\frac{y\lambda(1-v)}{Rp^2}$$

$$\frac{\sigma_{yz}}{\sigma_o} = b_x\left(\frac{v}{R} - \frac{y^2}{R^3}\right) + b_y\,\frac{xy}{R^3} - b_z\,\frac{x\lambda(1-v)}{R(R+\lambda)}, \qquad \text{(E-6)}$$

where $p^2 = x^2 + y^2$.

E-2: Stress Field About a Semi-Infinite Dislocation

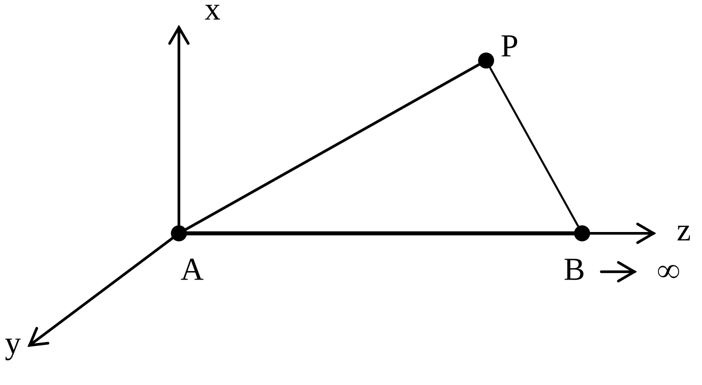


Figure E-2: Stress field around a semi-infinite segment

The stress field σ_{ij}^{P} at point P due to the semi-infinite dislocation AB is given by

$$\sigma_{ij}(x,y,z)^{P} = \sigma_{ij}(x,y,z)^{B} - \sigma_{ij}(x,y,z)^{A}, \qquad \text{(E-7)}$$

where $\sigma_{ij}(x,y,z)^{A}$ can be obtained by any set of equations listed previous pages and for $\sigma_{ij}(x,y,z)^{B}$:

$$\frac{\sigma_{xx}}{\sigma_o} = \frac{2b_x y}{x^2+y^2}\left(1+\frac{2x^2}{x^2+y^2}\right)+\frac{2b_y x}{x^2+y^2}\left(1-\frac{2x^2}{x^2+y^2}\right)$$

$$\frac{\sigma_{xy}}{\sigma_o} = \frac{2b_x x}{x^2+y^2}\left(1+\frac{2y^2}{x^2+y^2}\right)+\frac{2b_y y}{x^2+y^2}\left(1-\frac{2x^2}{x^2+y^2}\right)$$

$$\frac{\sigma_{yy}}{\sigma_o} = -\frac{2b_x y}{x^2+y^2}\left(1+\frac{2y^2}{x^2+y^2}\right)+\frac{2b_y x}{x^2+y^2}\left(1-\frac{2y^2}{x^2+y^2}\right)$$

$$\frac{\sigma_{xz}}{\sigma_o} = \frac{2b_z y(1-v)}{x^2+y2}$$

$$\frac{\sigma_{yz}}{\sigma_o} = -\frac{2b_z x(1-v)}{x^2+y2}$$

$$\frac{\sigma_{xz}}{\sigma_o} = \frac{4b_x vy}{x^2+y2}-\frac{4b_x vx}{x^2+y2} \qquad \text{(E-8)}$$

14. References

Demir, I., Hirth, J.P. and Zbib, H.M., "Remarks on the Stress Field of the Somigliana Ring Dislocation", *Mechanics Research Communications*, 19, 396-374, 1992.

Demir,I., Hirth,J.P. and Zbib,H.M., "The Extended Stress Field Around the Cylindrical Crack Using the Theory of Dislocation Pile-ups", Int. J. Engng. Sci., 30(7), 829-845, 1992.

Demir, I., Hirth, J.P. and Zbib, H.M., " The Somigliana Ring Dislocation" *J. Elasticity*, 28, 223-246, 1992.

Demir, I., Hirth, J.P and Zbib, H.M.,"Interaction Between Interfacial Ring Dislocations", *Int. J. Engrg. Sci.*, 31, 483-492, 1993.

Demir, I., Zbib, H.M. and Hirth, J.P "The Interfacial Somigliana Ring Dislocation", *Journal of the Mechanical Behavior of Materials*, 5,47-57, 1993.

Hirth J.P. and Lothe J., Theory of Dislocations, 2nd Edition, Krieger, Melbourne, (1992).

Hirth, J.P, Rhee, M. and Zbib, H.M., "Modeling of Deformation by a 3D Simulation of Multipole, Curved Dislocations", J. Computer-Aided Materials Design, 3, 164-166, 1996.

Hirth, J.P, Rhee, M. and Zbib, H.M., "Modeling of Deformation by a 3D Simulation of Multipole, Curved Dislocations", J. Computer-Aided Materials Design, 3, 164-166, 1996.

Hirth, J.P., Zbib, H.M. and Lothe J., "Forces on High Velocity Dislocations", Modeling & Simulations in Maters. Sci. & Enger, 6, 165-169, 1998.

Huang, H., Ghoniem, N., Diaz de la Rubia, T., Rhee, Zbib. H.M. and Hirth, J.P., "Development of Physical Rules for Short Range Interactions in BCC Crystals", in press ASME-JEMT, 1999

Khraishi, T.A., Zbib, H.M., Hirth, J.P. and Khaleel, M., "Analytical Solution for The Stress-Displacement Field of Glide Dislocation Loop", to be submitted to Int. J. Enger. Sci., 1998.

Rhee, M., Hirth, J.P. and Zbib, H.M., ``A Superdislocation Model for the Strengthening of Metal Matrix Composites and the Initiation and Propagation of Shear Bands", *Acta Metallurgica et Materialia*, 42, 2645-2655, 1994.

Rhee, M., Hirth, J.P. and Zbib, H.M., ``On the Bowed Out Tilt Wall Model of Flow Stress and Size Effects in Metal Matrix Composites", *Scripta Metall. et Materialia*, 31, 1321-1324, 1994.

Rhee, M., Zbib, H.M., Hirth, J.P., H. Huang and T. D. de La Rubia, "Models for Long/Short Range Interactions in 3D Dislocatoin Simulation", Modeling & Simulations in Maters. Sci. & Enger, 6, 467-492, 1998.

Zbib, H.M. and Demir. I.,``The Interface Ring Dislocation in Fiber-Matrix Composites: Approximate Analytical Solution", ASME *Journal of Engineering Materials and Technology*, 116, 279-285, 1994.

Zbib, H.M., Rhee, M. and Hirth, J.P., "On Plastic Deformation and the Dynamcis of 3D Dislocations", *Int. J. Mech. Science*, 40, 113-127, 1998.

Zbib, H.M., Rhee, M. and Hirth, J.P., *"3D Simulation of Curved Dislocations: Discretization and Long Range Interactions*", in: Advances in Engineering Plasticity and its Applications, eds. Abe and T, Tsuta. Pergamon, NY, 1996, pp. 15-20.

Zbib,H.M., de La Rubia, T. D., Rhee, Rhee and Hirth, J.P, "3D Dislocation Dynamics: Stress-Strain behavior and Hardening Mechanisms in FCC and BCC Metals. J. Nuc. Maters., 276, 154-165, 2000.

Zbib, H.M., Rhee, M. Hirth, J.P. and Diaz de la Rubia, T., "On Dislocation Reactions and Hardening Mechanisms in Dislocation Dynamics", Proceedings of the MRS 1998, in press.