

Das IT- und Medienstipendium für innovative Projekte von Studierenden

Daniel Schabunow

Hochschule Karlsruhe – Technik und Wirtschaft

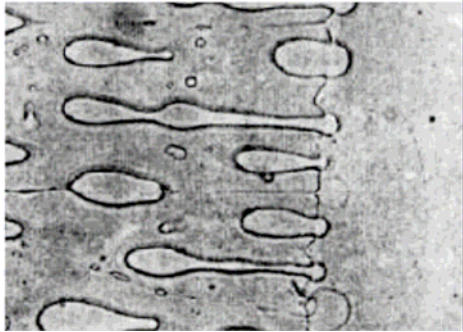
Design und Implementierung eines Strömungslösers zur numerischen Simulation von Kristallwachstum unter Einfluss von Konvektion

Abschlussbericht zum Stipendiumsprojekt
vom 01.01.2005-31.12.2005

Das Ziel des Projektes war die Entwicklung eines Strömungslösers zur numerischen Simulation von Kristallwachstum unter dem Einfluss von Konvektion. Hierzu wurde zuerst ein geeigneter Strömungslöser ausgewählt. Dieser wurde an den Kristallwachstumssimulator angekoppelt, der auf einem mathematisch-physikalischen Modell basiert und in der Forschergruppe entwickelt wurde. Das mathematisch-physikalische Modell wird ausführlich in [1] beschrieben.

An dieses Modell wurden die Datenstrukturen des Strömungslösers angepasst. Die simultane Lösung der Strömungsgleichungen (Navier-Stokes Gleichungen) und der Evolutionsgleichungen für das Kristallwachstum erforderte eine Modellerweiterung, das Design eines transparenten, modularen und gut strukturierten Codes und die Implementierung aller Kopplungskomponenten. Im Anschluss an die Programmentwicklung wurden Testsimulationen für einfache Materialsysteme und Rechenszenarien durchgeführt, bei denen Strömung in der flüssigen Phase (in der Metallschmelze) den Kristallisationsprozess beeinflusst.

Ein Forschungsgebiet der Materialwissen-



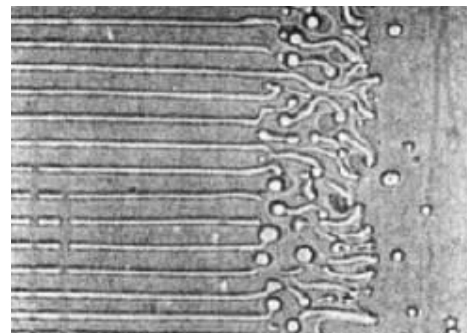
schaften beschäftigt sich mit der Evolution von Kristallen. Dabei ist unter anderem von Interesse, wie sich die Strukturen bei der Erstarrung von Metallschmelzen entwickeln. Dieser Vorgang ist jedoch nicht oder nur schwer in Experimenten zu beobachten. Aus diesem Grund wird versucht, diesen Vorgang so realistisch wie möglich in einer Simulation nachzubilden.

Abbildung 1: Beispiele von Monotektika aus Experimenten mit und ohne Konvektion. (Quelle [2])

Eines der Ziele während der Laufzeit des Karl-Steinbuch Stipendiums 2005 war es, Monotektika zu simulieren. Bei der Erstarrung einer Flüssigphase L_1 bilden sich eine neue Flüssigphase L_2 und eine Festphase S . Bedingt durch eine Mischungslücke wird die Phase L_2 nicht aufgelöst, sondern während des Erstarrungsprozesses eingeschlossen. Wie in der Abbildung 1 rechts zu sehen ist, wird durch den Einfluss von Konvektion (temperaturgetriebene Strömung) der Erstarrungsprozess erheblich beeinflusst.

Es gibt aber auch noch andere Erstarrungsvorgänge, die durch Konvektion beeinflusst werden. So werden bisher Dendriten (Seitenarme von Kristallen) simuliert, deren Wachstum den Bedingungen unter Schwerelosigkeit entsprechen. Wird zusätzlich Konvektion berücksichtigt,

ist eine Änderung der Kristallentwicklung zu erwarten. Es sollten sich unter Schwerkraft beobachtete Formen ausbilden. In der hier vorliegenden Arbeit soll gezeigt werden, wie ein Phasenfeldmodell zur Simulation von Kristallwachstum an ein mit Navier-Stokes Strömungsgleichungen beschriebenes Strömungsmodell angebunden wird. Nach der Vorstellung des Modells, das die Grundlage dieser Arbeit bildet, werden die Schritte dargelegt, die notwendig sind, die Gleichungen in diskreter Form darzustellen. Es wird erläutert, wie die Kopplung zweier bisher voneinander unabhängig entwickelter Löser durchgeführt wurde. Mit Hilfe des verwendeten Ansatzes sollen im Rahmen dieser Arbeit qualitative Ergebnisse gewonnen werden, die zeigen sollen, ob dieser gekoppelte Modellansatz als Basis für weiterführende Forschung in der Pha-



senfeldmodellierung dienen kann.

Projektverlauf

Am Anfang des Projektes standen zwei einzelne, nicht miteinander gekoppelte Löser zur Verfügung. Die Kopplung dieser beiden Einzellöser wurde schrittweise durchgeführt. Folgende Teilschritte waren dazu erforderlich:

1. Entwicklung eines mathematischen Modells zur Beschreibung von Erstarrungsvorgängen unter dem Einfluss von Strömung,

2. Anpassung eines vorhandenen Strömungslösers auf die Datenstrukturen des Kristallsimulators,
3. Kopplung der beiden Programme: Kristallwachstumslöser und Strömungslöser und Konzeptionierung des simulanten Programmablaufs,
4. Erste gekoppelte Simulationen mit fest-flüssig Phasengrenzen während eines Strömungsprozesses.

Ergebnisse

1. Entwicklung eines mathematischen Modells zur Beschreibung von Erstarrungsvorgängen unter dem Einfluss von Strömung

Die grundlegende Idee ist, den Kristall nicht als explizites Hindernis zu beschreiben, sondern als Flüssigkeit mit einer 50- bis 500-fach größeren Viskosität als die der Schmelze. Es wird deshalb ein raumabhängiger Viskositätsparameter $\mu(\phi)$ eingeführt, der im gesamten Gebiet berechnet und zur Erreichung der numerischen Stabilität im diffusen Interface mit einer von den zwei Mittelwertbildungen

$$\mu(\phi) = \sum_{\alpha=1}^N \mu_{\alpha} \phi_{\alpha} \quad (\text{arithmetisch})$$

oder

$$\frac{1}{\mu(\phi)} = \sum_{\alpha=1}^N \frac{\phi_{\alpha}}{\mu_{\alpha}} \quad (\text{harmonisch})$$

ermittelt wird. Dabei entspricht μ_{α} dem Wert der Viskosität in der Phase α .

Phasenfeldmodell:

Basierend auf dem in [2] formulierten Ansatz zur Einbindung von Strömung in die Beschreibung von Phasenübergängen wurde ein thermodynamisch konsistentes Multi-Phasenfeldmodell mit inkompres-

siblen Navier-Stokes Gleichungen entwickelt. Das Modell basiert auf folgendem Entropiefunktional innerhalb eines betrachteten Volumens $\Omega(t)$

$$S = \int_{\Omega(t)} \left[\rho s(\phi, c, T) - \left(\varepsilon a(\phi, \nabla \phi) + \frac{1}{\varepsilon} w(\phi) \right) \right] dV$$

wobei ρ die Dichte und s die Entropiedichte pro Masseneinheit ist. Der zweite Term in runden Klammern repräsentiert die Oberflächenanteile der Entropie und beinhaltet die diffuse Grenzflächenbreite ε , die Oberflächenentropiedichte in der Formulierung

$$a(\phi, \nabla \phi) = \sum_{\alpha < \beta}^N \frac{\gamma_{\alpha\beta}}{m_{\alpha\beta}} (A(q_{\alpha\beta})),$$

mit den verallgemeinerten Gradientenvektoren

$$\vec{q}_{\alpha\beta} = \phi_{\alpha} \nabla \phi_{\beta} - \phi_{\beta} \nabla \phi_{\alpha}$$

und das Multi-Well Potential $\omega(\phi)$, das fester Bestandteil des Phasenfeldmodells ist. Die physikalischen Gesetze der Massenerhaltung, Erhaltung der inneren Energie und der zweite Hauptsatz der Thermodynamik können als Evolutionsgleichungen in der folgenden Form geschrieben werden:

$$\omega \varepsilon \frac{D\phi_{\alpha}}{Dt} = \varepsilon \nabla \cdot (\rho a_{,\nabla \phi_{\alpha}}(\phi, \nabla \phi) - \rho a_{,\phi_{\alpha}}) - \frac{\rho}{\varepsilon} w_{,\phi_{\alpha}}(\phi) - \rho \frac{f_{,\phi_{\alpha}}}{T} - \lambda$$

$$\frac{Dc_i}{Dt} = -\nabla \cdot \left(L_{i0} \nabla \frac{1}{T} + \sum_{j=1}^K L_{ij} \nabla \left(\frac{-\mu_j}{T} \right) \right)$$

$$\frac{De}{Dt} = -\nabla \cdot \left(L_{00} \nabla \frac{1}{T} + \sum_{j=1}^K L_{0j} \nabla \left(\frac{-\mu_j}{T} \right) \right) + m : \nabla \vec{u}$$

Dabei handelt es sich bei $m = \sigma + \theta_{cap}$ und ∇u um Matrizen. Die Operation ':' bedeutet ein komponentenweises Multiplizieren und anschließendes Aufaddieren aller Terme.

Zusätzlich wird die Transportableitung $D/Dt = \partial/\partial t + \vec{u} \cdot \nabla$ eingeführt. Dabei ist zu erkennen, dass im Wesentlichen außer dem zusätzlichen advektiven Term zu den Gleichungen des Phasenfeldmodells keine weiteren Änderungen nach der Kopplung mit der Strömungsdynamik vorliegen. Zunächst werden ausschließlich isotherme Prozesse betrachtet. Eine ausführliche Darstellung des Phasenfeldmodells ist in [3] nachzulesen.

Navier-Stokes Gleichungen:

Für das Geschwindigkeitsfeld \vec{u} sind die Navier-Stokes Gleichungen gegeben mit

$$\rho \frac{D\vec{u}}{Dt} = \nabla \cdot (\sigma + \theta_{cap}) = \nabla \cdot \left[-p\mathbf{I} + \mu(\phi)(\nabla\vec{u} + (\nabla\vec{u})^T) + \theta_{cap} \right]$$

$$\nabla \cdot \vec{u} = 0.$$

Diese für die Kopplung modifizierten Gleichungen gelten für inkompressible Strömungen. Der Spannungstensor σ enthält als treibende Kraft der Flüssigkeitsströmung den Druck p (\mathbf{I} bezeichnet die Einheitsmatrix) und die innere Reibung, die proportional zur Viskosität $\mu(\phi)$ ist. Die kapillaren Reibungskräfte an den Grenzflächen werden durch den in der Navier-Stokes Gleichung neu eingeführten Kapillartensor θ_{cap} beschrieben. Dieser Tensor ist bei der Behandlung von Strukturen im Mikrobereich von Bedeutung.

Ein Bezug mit der Oberflächenentropiedichte im Phasenfeldmodell ergibt sich durch

$$\theta_{cap} = \left[a(\phi, \nabla\phi)\mathbf{I} - \sum_{\alpha=1}^N \left(\frac{\partial a(\phi, \nabla\phi)}{\partial(\nabla\phi_\alpha)} \otimes \nabla\phi_\alpha \right) \right],$$

wobei \otimes das Tensorprodukt ist und die Summation über alle definierten Phasen N des Rechengebietes erfolgt.

2. Anpassung eines vorhandenen Strömungslösers auf die Datenstrukturen des Kristallsimulators

Die Programmstruktur des Navier-Stokes-Strömungslösers (NaSt3D) wurde untersucht und mit der Struktur des Kristallwachstumslösers (Pace3D) verglichen. Danach wurden Parserfunktionen des Pace3D zum Einlesen einer Parameterdatei in den NaSt3D eingebaut. Dadurch wurde es möglich, eine einzige Parameterdatei für beide Programme zu verwenden. Da viele Parameter des NaSt3D-Strömungslösers im Quellcode fest verankert sind, mussten einige Routinen angepasst werden, so dass eine komfortable Konfiguration über eine Textdatei möglich ist.

Als nächstes wurde der Ablauf des Programms überarbeitet. Aufbauend auf der Pace3D-Programmstruktur, die in C programmiert und objektorientiert ist, wurden die einzelnen Randbedingungen aus verschachtelten If-Abfragen in eigene Klassen übertragen. Zusätzlich wurden auch die Initialisierungsroutinen des restlichen Lösers aufgeteilt und den entsprechenden Klassen zugeordnet. Diese Routinen kann man mit den Konstruktoren in C++ vergleichen. Die Speicherroutinen für Datensätze speichern alle im NaSt3D berechneten Variablen in einer einzigen Datei ab. Einige dieser Variablen finden zurzeit keine Verwendung und wurden deswegen entfernt. Von Interesse sind lediglich die Werte für die Geschwindigkeit und für den Druck.

Vorbereitung des Kristallwachstumslösers (PACE3D):

Der Pace3D-Löser ist die Basis für den gekoppelten Löser. Daher musste keine Neustrukturierung des Programms durchgeführt werden. In der Verzeichnisstruktur des Pace3D's wurden in einem Unterverzeichnis 'nast3d' die NaSt3D-Quelldateien abgelegt und in den vorhandenen Übersetzungsprozess aufgenommen.

Die Hauptfunktion des NaSt3D wurde in eine eigene Klasse 'nast3d', die sich um die Initialisierung, Ausführung und das Aufräumen des NaSt3D-Lösers kümmert, überführt und zur Kopplung in den Pace3D-Teil integriert. Die Initialisierung des NaSt3D wurde in die Hauptklasse 'solver' des Pace3D-Teils eingebaut. Die Durchführung der eigentlichen Simulation wird in der Funktion 'execSolver' gestartet. Hier wird die als Parameter angegebene Zeitschrittzahl in einer Schleife berechnet.

In dieser Schleife wird nun auch die neue Funktion 'execNaSt3D' aufgerufen. Die bisherigen Initialisierungsroutinen der ursprünglichen Löser verwendeten noch von einander unabhängige Parameter und auch Variablen zu deren Verwaltung. Um diese aneinander anzugleichen, wurden nun im NaSt3D-Teil alle Variablen, die in beiden Löserteilen vorkommen, auf die im Pace3D-Teil verwendeten umgestellt. Dabei handelte es sich vor allem um Variablen der Parameter, die das Gebiet beschreiben, wie die Anzahl der Zellen in x-, y- und z-Richtung (N_x , N_y und N_z), die entsprechenden Abstände ∂x , ∂y und ∂z und die Zeitschrittweite ∂t .

Nun war es möglich, durch einen Aufruf des Pace3D-Programms beide Löser zu starten. Dabei kann über die Parameterdatei ausgewählt werden, welche Gleichungen des neuen Lösers ausgeführt werden sollen. Damit waren die Vorbereitungen im Pace3D abgeschlossen, und es konnte mit der Kopplung des Modells begonnen werden.

3. Kopplung der beiden Programme: Kristallwachstumslöser und Strömungslöser und Konzeptionierung des Programmablaufs

In diesem Stadium der Entwicklung lag ein gekoppeltes Programm vor, das zwei unabhängige Modelle löste. Diese mussten nun Schritt für Schritt zusammengeführt werden. Als erstes wurde die diskrete Form des viskosen Spannungstensors die Navier-Stokes Gleichung in den NaSt3D-Teil eingebaut. Dadurch war es jetzt möglich, verschiedene Phasen mit unterschiedlichen Viskositäten zu simulieren.

Zur vollständigen Umsetzung der neuen Terme der Navier-Stokes Gleichung wurde der Kapillartensor implementiert. Dieser benötigt die Oberflächenentropiegleichung, daher wurden die Funktionen 'calcThetaU', 'calcThetaV' und 'calcThetaW' für jede Komponente (u , v , w) des Geschwindigkeitsvektors $\vec{u}_{i,j,k}$ neu entwickelt und in den Pace3D-Teil 'function_a.c' integriert. Zur Berechnung der linksseitigen Divergenz wird der jeweilige Teil des Tensors an der Stelle (i, j, k) und an der Stelle des nächst niedrigeren Werts in x-, y- bzw. z-Richtung, d.h. $(i-1, j, k)$, $(i, j-1, k)$ bzw. $(i, j, k-1)$, benötigt. Dies geschieht durch zweimaliges Aufrufen der zugehörigen Funktionen.

Nun wurde noch der neu hinzu gekommene advective Term aus der Phasenfeldgleichung übernommen. Es mussten nur wenige Änderungen durchgeführt werden, da die zentralen Gradienten von ϕ bereits berechnet in einer Pufferschicht vorliegen. Lediglich die Geschwindigkeit musste gemittelt werden, damit sie auf der Position von $\phi_{i,j,k}$ liegt. Die angepasste Konzentrationsgleichung benötigt zentrale Gradienten, die bisher noch nicht berechnet wurden. Daher war es nötig, in der Pufferschicht auch die Berechnung dieser Gradienten zu integrieren. Ansonsten musste nur die Mittelung der Geschwindigkeit berücksichtigt werden. Die Temperaturgleichung wurde in dieser Arbeit nicht angepasst und deswegen im NaSt3D deaktiviert, da die gekoppelte Energiegleichung später im PACE3D durch die dort zu modifizierende Temperaturgleichung berechnet werden soll.

Struktur und Ablauf des Programm-Codes:

Die Durchführung einer Simulation beginnt mit der Erstellung einer Parameterdatei, darin sind die für die Berechnung notwendigen Parameter Gebietsgröße (N_x, N_y, N_z), Zellabstände ($\partial x, \partial y, \partial z$) und Zeitschrittweite (∂t_n) festgelegt. Das Gebiet kann mit Phasenfeld-, Konzentrations-, Druck- und Geschwindigkeitswerten (ϕ_α, c_i, p und \vec{u}) gefüllt werden. Weiterhin werden die Gleichungen, die zu berechnen sind, und die darin verwendeten Konstanten angegeben. Für die Gebietsränder muss man auch Randbedingungen für jede Größe bestimmen. Zum Starten des Löser werden die Parameterdatei und die Basisnamen für die Ausgabedateien auf die Kommandozeile übergeben. Anhand der

in der Parameterdatei definierten Parameter initialisiert der Löser das Gebiet und die Gleichungen.

In der Abb. 2 ist der Programmablauf für gleichzeitig aktivierte Phasenfeld- und Navier-Stokes Gleichungen in Form eines Flussdiagramms dargestellt. Die Zeitschrittweite ∂t_n wird als konstanter Wert vorgegeben. Sie muss das Stabilitätskriterium des Phasenfeldmodells erfüllen. Die Zeitschrittweite ∂t_{nNaSt} wird zur numerischen Stabilität vor jeder Iteration über die Navier-Stokes Gleichungen neu berechnet. Falls das neu berechnete ∂t_{nNaSt} zur Überschreitung von ∂t_{n+1} führt, wird es vom Algorithmus reduziert, so dass ∂t_{n+1} exakt erreicht wird. Frühe Experimente haben gezeigt, dass das ∂t_n , das für eine stabile Lösung der Gleichungen des Phasenfeldmodells benötigt wird, größer ist als ∂t_{nNaSt} . Somit wird mehrfach über die Navier-Stokes Gleichungen iteriert bis der nächste Zeitschritt ∂t_n erreicht wird. Dies hat einen großen Einfluss auf die Rechenzeit.

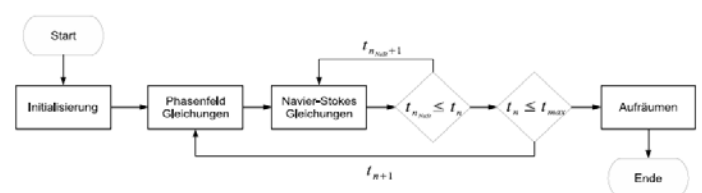


Abb. 2: Flussdiagramm des gekoppelten Phasenfeld- und Navier-Stokes-Lösers. Der in dem Flussdiagramm in Abb. 2 auftretende Index n kennzeichnet den n -ten Zeitschritt t_n der Berechnung. Das Programm endet mit dem Erreichen des Zeitschritts t_{\max} .

Schematisch wird für die ersten drei Zeitschritte gezeigt, welche Werte aus den

einzelnen Zeitschritten von den Gleichungen verwendet werden:

$$\phi_0(\phi_{init}, \vec{u}_{init})$$

$$\vec{u}_0(\vec{u}_{init}, \phi_0)$$

$$\phi_1(\phi_0, \vec{u}_0)$$

$$\vec{u}_1(\vec{u}_0, \phi_1)$$

$$\phi_2(\phi_1, \vec{u}_1)$$

$$\vec{u}_2(\vec{u}_1, \phi_2)$$

Die berechneten Werte sind dargestellt durch \vec{u} für den Geschwindigkeitsvektor der Navier-Stokes Gleichungen und durch ϕ für die Phasenzustandsvariable der Phasenfeld-Gleichung. In Klammern stehen die benötigten Werte zur Berechnung des neuen Ergebnisses.

4. Erste gekoppelte Simulationen mit fest-flüssig Phasengrenzen während eines Strömungsprozess

In der durchgeführten Simulation wurde die Phasenfeldgleichung aktiviert und die Temperatur auf einen konstanten, unterkühlten Wert gesetzt, so dass die feste Phase im Gleichgewicht ist und weder wächst noch schmilzt, d. h. in ihrem Volumen völlig konstant bleibt. Abb. 3 zeigt, dass die anfangs kreisförmig gesetzte feste Phase sich auf Grund der von links nach rechts laufenden Strömung verformt. In den weiteren Schritten bewegt sich das Partikel, getrieben durch die Strömung, zum rechten Rand hin. Die Bilder links stellen den Bereich des festen Partikels und der Phasengrenze dar, während die Bilder rechts das Strömungsfeld visualisieren.

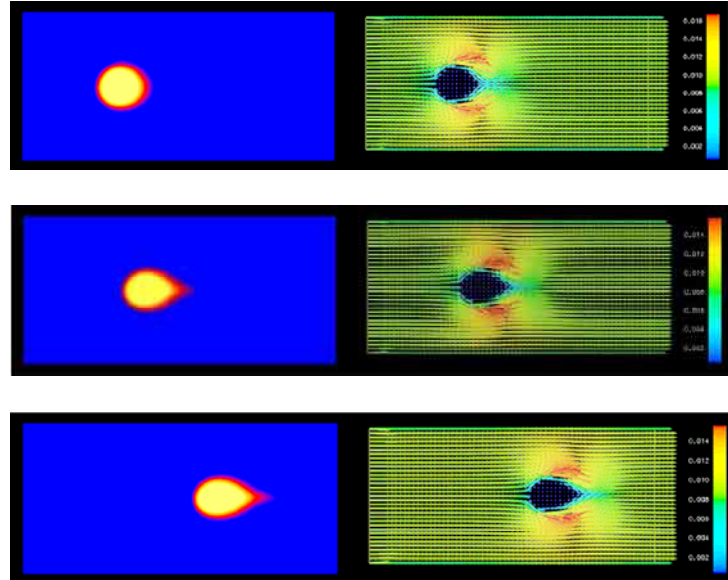


Abb. 3: Eine in ein Strömungsfeld kreisförmig eingesetzte, feste Phase verändert ihre Form und ihre Position in der Flüssigphase unter dem Einfluss von Strömung, die in der rechten Bildspalte als Pfeillinien visualisiert wurde.

Ausblick

Basierend auf den Ergebnissen des Karl-Steinbuch-Stipendiums lassen sich die folgenden Anschlussprojekte und Arbeiten definieren:

- Auf Grund des hohen Rechenzeitbedarfs bei der gekoppelten Behandlung von Kristallwachstum und Strömung sind Optimierungen des Quellcodes erforderlich. Für die Rechenzeit aufwendigen Simulationen sollen die gekoppelten Löser mit Hilfe von OpenMP und MPI parallelisiert und auf einem Mehrprozessorcluster berechnet werden. Die Techniken OpenMP und MPI werden bereits im Pace3D-Kristallwachstumslöser verwendet.

- Um die Strömung durch verschiedene Methoden (Vektorpfeile, Streamlines, etc.) visualisieren zu können, sollen die im Projekt verwendeten Visualisierungstools erweitert und insbesondere für die Darstellung in drei Dimensionen vorbereitet werden.
- Es soll untersucht werden, wie sich der Einfluss von Strömung auf die Erstarrungsprozesse vor allem in eutektischen und monotektischen Legierungen auswirkt.

Veröffentlichung des Stipendiaten

Nestler, F. Wendler, T. Frodl, D. Schabunow;
Ein paralleler 3D Simulator zum mikrostrukturbasierten Materialdesign,
Fraunhofer IRB Verlag, Herausgeber: Dieter Spath, Klaus Haasis, Dieter Klumpp,
Aktuelle Trends in der Softwareforschung, ISBN 3-8167-6955-1 (2005), 239-258.

Literaturverzeichnis

[1] H. Garcke, B. Nestler, B. Stinner, A diffuse interface model for alloys with

multiple components and phases, SIAM J. Appl. Math. 64, 775 ff. (2004)

[2] B. Nestler, A. A. Wheeler, L. Ratke, C. Stoecker: Phase-field model for solidification of a monotectic alloy with convection. Physica D 141, (2000) 133 - 154.

[3] M. Selzer, Diplomarbeit (2005).

Anlagenverzeichnis

- Veröffentlichung des Stipendiaten: Nestler, F. Wendler, T. Frodl, D. Schabunow;
Ein paralleler 3D Simulator zum mikrostrukturbasierten Materialdesign, Fraunhofer IRB Verlag, Herausgeber: Dieter Spath, Klaus Haasis, Dieter Klumpp, Aktuelle Trends in der Softwareforschung, ISBN 3-8167-6955-1 (2005), 239-258.
- Bild des Deckblatts: deckblattbild.png
- Abb. 1.: Experimentelle Bilder: abb1-links.png und abb1-rechts.png
- Abb. 2.: Flussdiagramm: abb2.png
- Abb. 3. Simulation: abb3-oben.png und abb3-mitte-unten.png

Karl-Steinbuch-Stipendium

Karl-Steinbuch-Stipendium
Breitscheidstr. 4
70174 Stuttgart
Tel. +49/711/90715/314
stiftung@mfg.de

Über das Stipendiumprogramm

Mit dem Karl-Steinbuch-Stipendium fördert die MFG Stiftung
Baden-Württemberg innovative wissenschaftliche und
künstlerische IT- und Medienprojekte, die Studierende aus
Baden-Württemberg zusätzlich zu Ihrem Studium durchführen.
Die Projekte dauern 6-12 Monate und werden mit bis zu
9.600 € gefördert.

Weitere Informationen:

www.karl-steinbuch-stipendium.de