

MatForge: Collaborative Computational Materials Research & Education

UMC Workshop on Computational Materials Education
Northwestern University, June 23-24, 2010

Laura Bartolo
Kent State University

Outline

- Introduction
- Open source
- Community building
 - Partnerships & Workshops
 - Hosting/Mirroring
- Research-based curriculum aids
- Easy authoring and broad dissemination
- Participating in MatForge
- Concluding remarks

MATFORGE

Part of the Materials Digital Library Pathway
MATDL

What is MatForge (<http://matforge.org>) ?

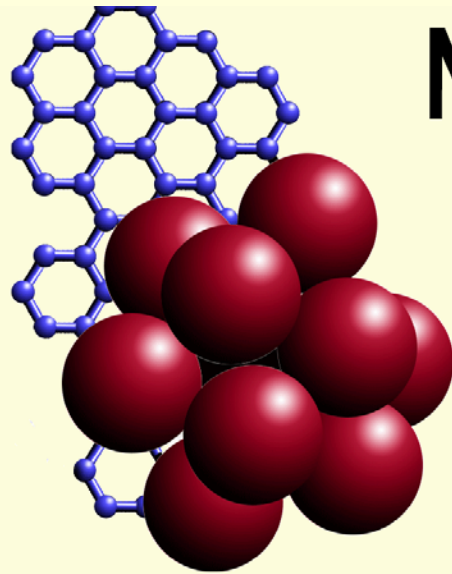
- **Provides** a branded, trusted, non-commercial, and neutral host/mirror site supporting materials code development.
- **Promotes** interaction between materials research, code development, and education by supporting:
 - materials code developers;
 - materials community to use code; and
 - **faculty to integrate code & aids into their courses.**

Open Source

- **MatForge platform:** Choice of software repositories with a web interface - Redmine
 - Subversion, Git, Mercurial, Bazaar – manage source code (version control, downloading services, and privacy control)
 - Redmine – communication tools
 - Wiki
 - Issue tracker
 - Interface to code repository
 - Forum
- **Code projects:**
 - Open source at your discretion
 - Effective instructional vehicle
 - Ways to work with proprietary codes

Community Building

A Community Platform supports synergy to:



Part of the Materials Digital Library Pathway
MATFORGE

As part of the NSF NSDL, MatDL Pathway provides a branded, trusted, non-commercial, and neutral site supporting open source, collaborative, materials code development.

Projects:

- NIST MSEL CTCMS
 - FiPy
 - Teaching with FiPy
- Carnegie Mellon Computational Materials Science
 - Mesoscale Microstructure Simulation Project (MMSP)
 - Microstructure Builder (coming soon)
 - Parallel Grain Growth 3D (PGG-3D)
 - Texture subroutines
- Department of Energy Computational Materials Science Network Cooperative Research Team
 - Dynamics and Cohesion of Materials Interfaces and Confined Phases Under Stress
- FDA Computational Materials Science
 - TheraPy
- Gibbs
- Lab for Computational Nanoscience and Soft Matter Simulation
 - GlotZilla
 - Image Processing GlotZilla (IPGZ)
 - Transition Path Sampling (TPS)
- Powell Research Group
 - Julian
 - RheoPlast
 - Illuminator
 - EBaporate
 - Ternary
- Thornton Research Group, University of Michigan
 - Li Battery cathode development (coming soon)
 - Topological Analysis with Level Set Method (coming soon)
 - Phase Field Modeling of Lipid Membranes (coming soon)
 - Simulation of Solid Oxide Fuel Cell Electrodes with Complex Microstructure (coming soon)
- Voorhees Research Group, Northwestern University
 - VLS-Growth (coming soon)

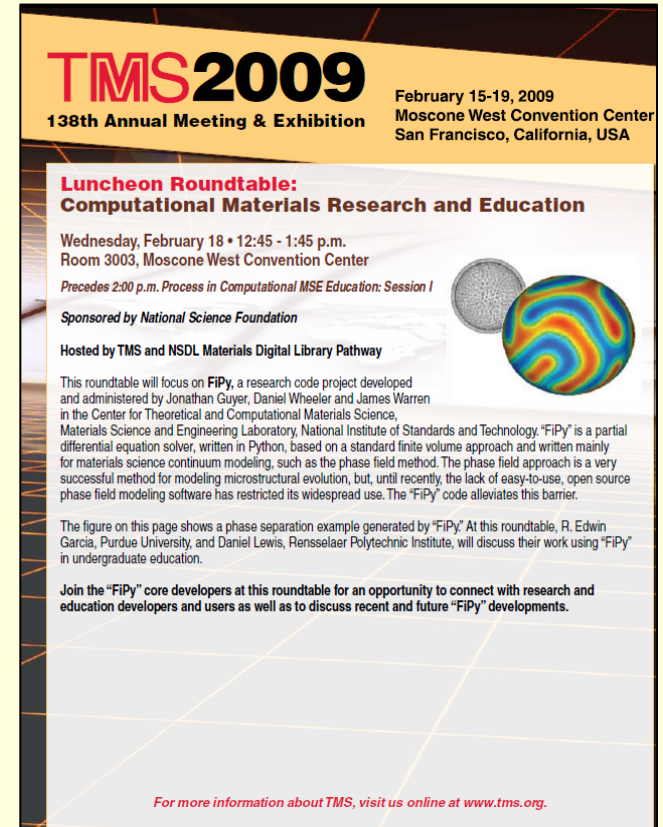
MatForge Live CDs:

- FiPy (Download Now, 655MB)
- Ternary (Download Now, 648MB)

- Promote U.S. innovation & competitiveness in development & use of materials
- Develop a pool of users ready to transfer the technology to industry, academia, and government labs
- Train next generation of scientists to use and develop computational materials tools

Community building with Materials Partners: TMS & ICME Committee

- TMS Committees
 - CoChair, ICME Education
 - Member, Education
- Education Symposia
- Roundtables
 - Spotlight materials codes
 - Discuss development & testing of curriculum aids
- MS&T 2010 TMS/MatDL
 - Oct 19-20, noon time



TMS2009
138th Annual Meeting & Exhibition

February 15-19, 2009
Moscone West Convention Center
San Francisco, California, USA

**Luncheon Roundtable:
Computational Materials Research and Education**

Wednesday, February 18 • 12:45 - 1:45 p.m.
Room 3003, Moscone West Convention Center

Precedes 2:00 p.m. Process in Computational MSE Education: Session I

Sponsored by National Science Foundation

Hosted by TMS and NSDL Materials Digital Library Pathway

This roundtable will focus on **FIPy**, a research code project developed and administered by Jonathan Guyer, Daniel Wheeler and James Warren in the Center for Theoretical and Computational Materials Science, Materials Science and Engineering Laboratory, National Institute of Standards and Technology. "FIPy" is a partial differential equation solver, written in Python, based on a standard finite volume approach and written mainly for materials science continuum modeling, such as the phase field method. The phase field approach is a very successful method for modeling microstructural evolution, but, until recently, the lack of easy-to-use, open source phase field modeling software has restricted its widespread use. The "FIPy" code alleviates this barrier.

The figure on this page shows a phase separation example generated by "FIPy." At this roundtable, R. Edwin Garcia, Purdue University, and Daniel Lewis, Rensselaer Polytechnic Institute, will discuss their work using "FIPy" in undergraduate education.

Join the "FIPy" core developers at this roundtable for an opportunity to connect with research and education developers and users as well as to discuss recent and future "FIPy" developments.

For more information about TMS, visit us online at www.tms.org.

Community building with Materials Partners: MRS & AAC

- MRS Committee
 - Collaborate with Academic Affairs
- Education Symposia
 - Presented & Chaired
- Workshops
 - Brief presentations
 - Curriculum discussions



MatDL & MRS Co-host Roundtable at 2010 MRS Annual Meeting

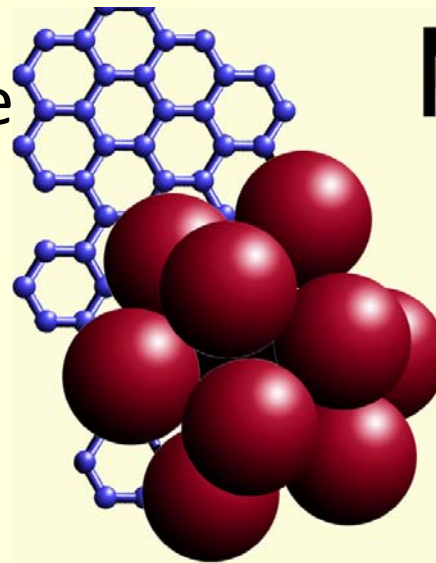
Monday, April 5

2:00 p.m. Demonstration | 3:30 p.m. Workshop
Marriott Marquis, 2nd Floor, Foothill G

Come preview the MRS/Materials Digital Library (MatDL) Expert Wiki & Planning Workshop on Translating Materials Research to Education! The roundtable presentation, sponsored by the National Science Foundation (NSF) and endorsed by the MRS Academic Affairs Committee (AAC), will be given by MatDL partners from the Massachusetts Institute of Technology (MIT), Carnegie Mellon University (CMU), and Kent State University (KSU). The preview demonstration and workshop will focus on starting development of the MRS/MatDL Expert Wiki, spotlighting materials advances to recruit undergraduates and high school students as well as testing Virtual Labs (VLs) in introductory science courses.

Community building by Hosting & Mirroring

- Hosting:
 - Develop on your machines, tools & timetable
 - Check in your code changes
 - Roll your code back
 - back up daily
 - Communicate over encrypted channels



Part of the Materials Digital Library Pathway
MATFORGE
MATDL

As part of the NSF NSDL, MatDL Pathway provides a branded, trusted, non-commercial, and neutral site supporting open source, collaborative, materials code development.

Projects:

- NIST MSEL CTCMS
 - FiPy
 - Teaching with FiPy
- Carnegie Mellon Computational Materials Science
 - Mesoscale Microstructure Simulation Project (MMSP)
 - Microstructure Builder (coming soon)
 - Parallel Grain Growth 3D (PGG-3D)
 - Texture subroutines
- Department of Energy Computational Materials Science Network Cooperative Research Team
 - Dynamics and Cohesion of Materials Interfaces and Confined Phases Under Stress
- FDA Computational Materials Science
 - TheraPy
- Gibbs
- Lab for Computational Nanoscience and Soft Matter Simulation
 - GlotZilla
 - Image Processing GlotZilla (IPGZ)
 - Transition Path Sampling (TPS)
- Powell Research Group
 - Julian
 - RheoPlast
 - Illuminator
 - EBaporate
 - Ternary
- Thornton Research Group, University of Michigan
 - Li Battery cathode development (coming soon)
 - Topological Analysis with Level Set Method (coming soon)
 - Phase Field Modeling of Lipid Membranes (coming soon)
 - Simulation of Solid Oxide Fuel Cell Electrodes with Complex Microstructure (coming soon)
- Voorhees Research Group, Northwestern University
 - VLS-Growth (coming soon)

MatForge Live CDs:

- FiPy (Download Now, 625MB)
- Ternary (Download Now, 645MB)

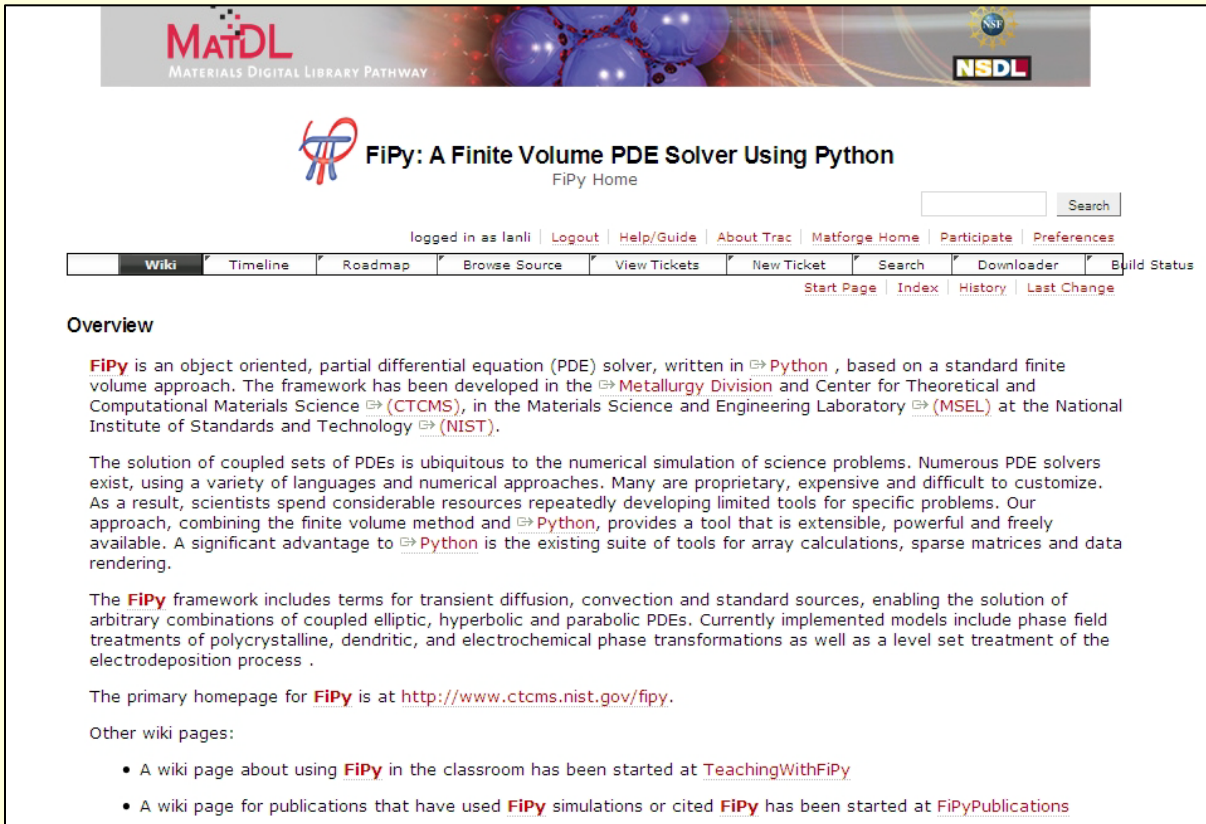
- Mirroring

Developing Research-based Curriculum Aids

Example: FiPy Research Code

• *Guyer, Wheeler, Warren, NIST*

A partial differential equation (PDE) solver, written in Python, based on a standard finite approach.



Overview

FiPy is an object oriented, partial differential equation (PDE) solver, written in [Python](#), based on a standard finite volume approach. The framework has been developed in the [Metallurgy Division](#) and Center for Theoretical and Computational Materials Science [\(CTCMS\)](#), in the Materials Science and Engineering Laboratory [\(MSEL\)](#) at the National Institute of Standards and Technology [\(NIST\)](#).

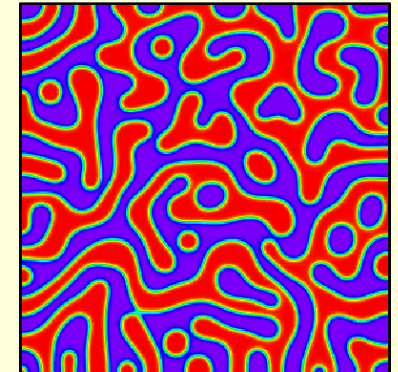
The solution of coupled sets of PDEs is ubiquitous to the numerical simulation of science problems. Numerous PDE solvers exist, using a variety of languages and numerical approaches. Many are proprietary, expensive and difficult to customize. As a result, scientists spend considerable resources repeatedly developing limited tools for specific problems. Our approach, combining the finite volume method and [Python](#), provides a tool that is extensible, powerful and freely available. A significant advantage to [Python](#) is the existing suite of tools for array calculations, sparse matrices and data rendering.

The **FiPy** framework includes terms for transient diffusion, convection and standard sources, enabling the solution of arbitrary combinations of coupled elliptic, hyperbolic and parabolic PDEs. Currently implemented models include phase field treatments of polycrystalline, dendritic, and electrochemical phase transformations as well as a level set treatment of the electrodeposition process.

The primary homepage for **FiPy** is at <http://www.ctcms.nist.gov/fipy>.

Other wiki pages:

- A wiki page about using **FiPy** in the classroom has been started at [TeachingWithFiPy](#)
- A wiki page for publications that have used **FiPy** simulations or cited **FiPy** has been started at [FiPyPublications](#)

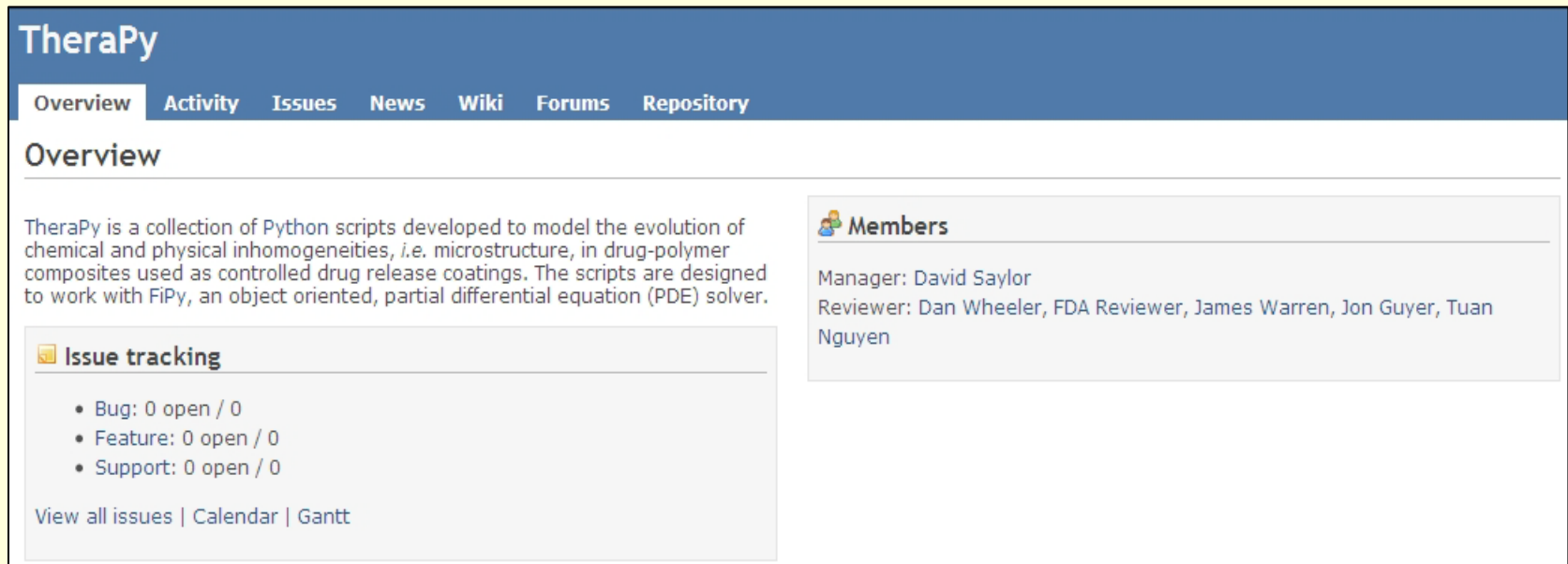


Cahn-Hilliard

Developing Research-based Curriculum Aids

- Using FiPy in research, *David Saylor, Food & Drug Administration*

TheraPy: a collection of Python scripts, designed to work with FiPy, developed to model the evolution of chemical and physical inhomogeneities in drug-polymer composites used as controlled drug release coatings.



The screenshot shows the GitHub repository page for 'TheraPy'. The page has a blue header with the project name 'TheraPy' and navigation tabs for 'Overview', 'Activity', 'Issues', 'News', 'Wiki', 'Forums', and 'Repository'. The 'Overview' tab is selected. Below the header, there is a description of the project: 'TheraPy is a collection of Python scripts developed to model the evolution of chemical and physical inhomogeneities, i.e. microstructure, in drug-polymer composites used as controlled drug release coatings. The scripts are designed to work with FiPy, an object oriented, partial differential equation (PDE) solver.' To the right of the description is a 'Members' section listing 'Manager: David Saylor' and 'Reviewer: Dan Wheeler, FDA Reviewer, James Warren, Jon Guyer, Tuan Nguyen'. Below the description is an 'Issue tracking' section with a list of issues: 'Bug: 0 open / 0', 'Feature: 0 open / 0', and 'Support: 0 open / 0'. At the bottom of the issue tracking section are links for 'View all issues | Calendar | Gantt'.

TheraPy

Overview Activity Issues News Wiki Forums Repository

Overview

TheraPy is a collection of Python scripts developed to model the evolution of chemical and physical inhomogeneities, *i.e.* microstructure, in drug-polymer composites used as controlled drug release coatings. The scripts are designed to work with FiPy, an object oriented, partial differential equation (PDE) solver.

Issue tracking

- Bug: 0 open / 0
- Feature: 0 open / 0
- Support: 0 open / 0

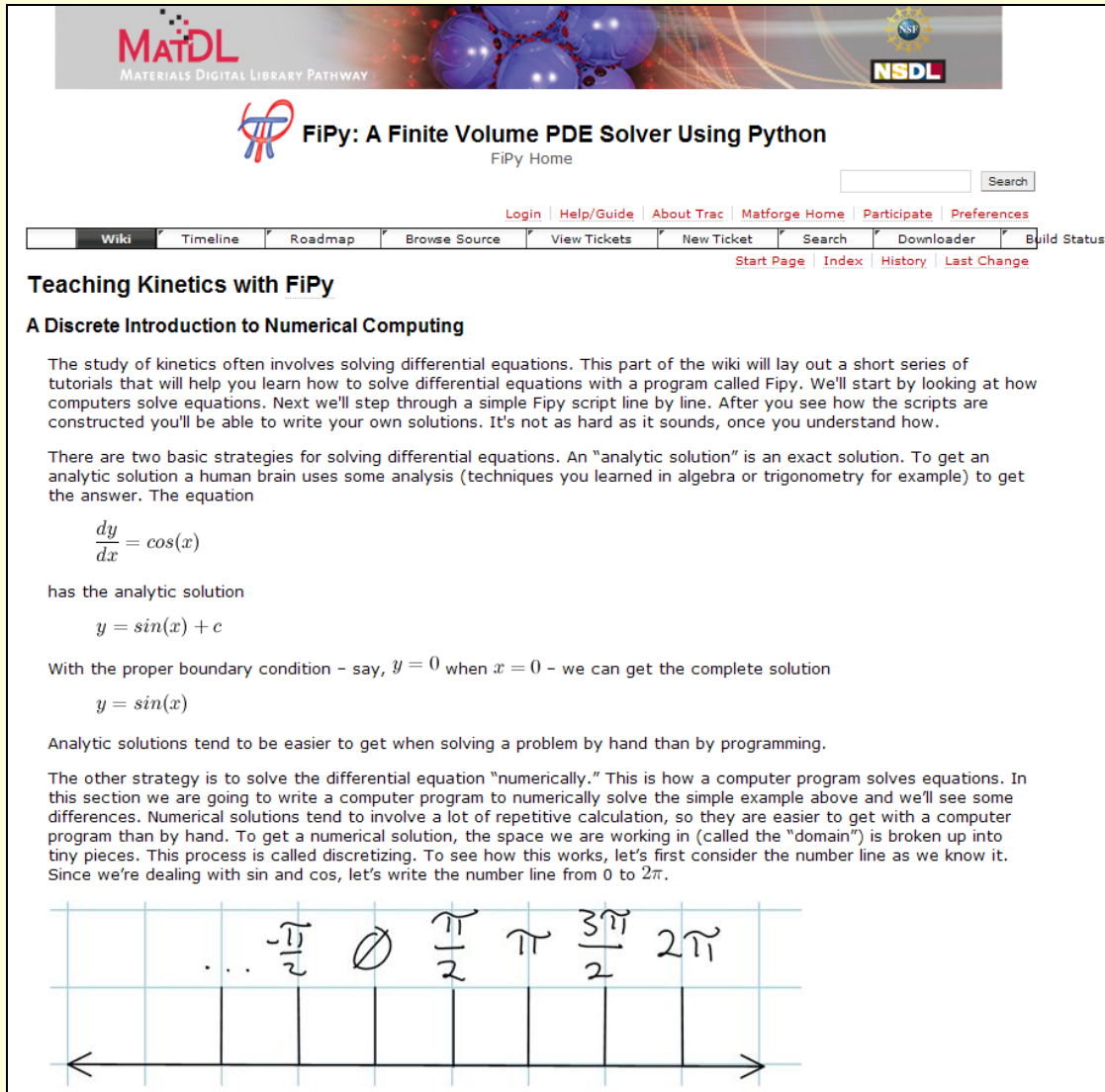
[View all issues](#) | [Calendar](#) | [Gantt](#)

Members

Manager: David Saylor
Reviewer: Dan Wheeler, FDA Reviewer, James Warren, Jon Guyer, Tuan Nguyen

Developing Research-based Curriculum Aids

- Teaching with FiPy



The screenshot shows the FiPy website interface. At the top, there are logos for MATDL (Materials Digital Library Pathway) and NSDL. Below the logos is the title "FiPy: A Finite Volume PDE Solver Using Python" and "FiPy Home". A search bar is present. A navigation menu includes links for Wiki, Timeline, Roadmap, Browse Source, View Tickets, New Ticket, Search, Downloader, and Build Status. Below the menu, there are links for Start Page, Index, History, and Last Change.

Teaching Kinetics with FiPy

A Discrete Introduction to Numerical Computing

The study of kinetics often involves solving differential equations. This part of the wiki will lay out a short series of tutorials that will help you learn how to solve differential equations with a program called FiPy. We'll start by looking at how computers solve equations. Next we'll step through a simple FiPy script line by line. After you see how the scripts are constructed you'll be able to write your own solutions. It's not as hard as it sounds, once you understand how.

There are two basic strategies for solving differential equations. An "analytic solution" is an exact solution. To get an analytic solution a human brain uses some analysis (techniques you learned in algebra or trigonometry for example) to get the answer. The equation

$$\frac{dy}{dx} = \cos(x)$$

has the analytic solution

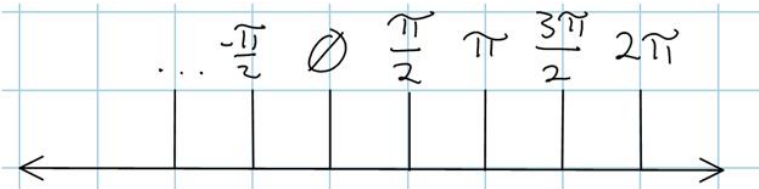
$$y = \sin(x) + c$$

With the proper boundary condition - say, $y = 0$ when $x = 0$ - we can get the complete solution

$$y = \sin(x)$$

Analytic solutions tend to be easier to get when solving a problem by hand than by programming.

The other strategy is to solve the differential equation "numerically." This is how a computer program solves equations. In this section we are going to write a computer program to numerically solve the simple example above and we'll see some differences. Numerical solutions tend to involve a lot of repetitive calculation, so they are easier to get with a computer program than by hand. To get a numerical solution, the space we are working in (called the "domain") is broken up into tiny pieces. This process is called discretizing. To see how this works, let's first consider the number line as we know it. Since we're dealing with sin and cos, let's write the number line from 0 to 2π .



A hand-drawn number line on a grid background. The line is horizontal and has arrows at both ends. Tick marks are drawn at regular intervals. Above the line, the following values are written from left to right: $-\frac{\pi}{2}$, 0 , $\frac{\pi}{2}$, π , $\frac{3\pi}{2}$, and 2π . Ellipses are drawn to the left of $-\frac{\pi}{2}$ and to the right of 2π .

FiPy Workbook:
Teaching Kinetics with
FiPy
Daniel Lewis, RPI

Easy Authoring and Broad Dissemination Code repository

Overview Activity Issues News Wiki Forums **Repository**

root / trunk Stat

Name	Size	Revision	Age	Author	Comment
▫ _build		12	about 1 month	David Saylor	updated links and fixed typos in scripts
▫ _static		1	2 months	David Saylor	initial import
▫ _templates		1	2 months	David Saylor	initial import
▫ scripts		12	about 1 month	David Saylor	updated links and fixed typos in scripts
▫ Makefile	2.9	1	2 months	David Saylor	initial import
▫ README.txt	4.4	12	about 1 month	David Saylor	updated links and fixed typos in scripts
▫ THEORY.txt	7.4	4	about 1 month	David Saylor	updated the documentation
▫ conf.py	6.3	1	2 months	David Saylor	initial import
▫ index.txt	1.4	8	about 1 month	David Saylor	

Latest revisions

#	Date	Author	Comment
12	05/13/2010 11:16 am	David Saylor	updated links and fixed typos in scripts
11	05/13/2010 10:57 am	David Saylor	created trunk
10	04/29/2010 12:38 pm	David Saylor	accidentally deleted some files
8	04/29/2010 12:31 pm	David Saylor	
4	04/27/2010 09:36 am	David Saylor	updated the documentation
2	03/25/2010 10:07 am	David Saylor	Updated theory section
1	03/24/2010 03:52 pm	David Saylor	initial import

[View differences](#)

[View revisions](#)

Powered by Redmine © 2006-2010 Jean-Philippe Lang

Easy Authoring and Broad Dissemination

Wiki

TheraPy

Overview Activity Issues News **Wiki** Forums Repository

TheraPy

History

Overview

TheraPy is a collection of Python scripts developed to model the evolution of chemical and physical inhomogeneities, *i.e.* microstructure, in drug-polymer composites used as controlled drug release coatings. This includes the development of microstructure during the process of casting drug eluting coatings, as well as the microstructural changes that occur during use or after the coating is immersed in the release medium. Therefore, these scripts provide tools to predict the relationships between materials and process variables, microstructure, and performance. Establishing these relationships can reduce empiricism in materials selection and process design, providing a facile and efficient means to tailor the underlying microstructure and achieve a desired drug release behavior.

The TheraPy scripts were developed in the Division of Chemistry and Materials Science (DCMS) in the Office of Engineering Laboratories (OSEL) at the Food and Drug Administration's (FDA) Center for Devices and Radiological Health (CDRH). The scripts are designed to work with FiPy, an object oriented, partial differential equation (PDE) solver. As described below, these scripts can be used much in the same way as the examples provided with the FiPy distribution.

Currently, scripts are provided to predict the microstructure development during casting in both 1D and 2D of systems composed of three components: drug, polymer, and solvent, as well as drug release behavior in 2D. In the future, we intend to add scripts that extend the current model to material systems that:

- are 3D
- contain arbitrary number of components
- exhibit crystallization
- form chemical bonds between constituents
- contain components that may undergo chemical degradation.

Download

A tarball of the TheraPy scripts and documentation can be downloaded from [export:release/therapy0.1.tar.gz](https://export.release.therapy0.1.tar.gz). Alternatively, you can checkout the latest version from the repository:

```
svn checkout https://matforge.org/svn/therapy/trunk therapy
```

To browse the source code on-line, click [here](#).

Documentation

Online versions of detailed documentation including the underlying theory and specific usage instructions can be found [here](#) (html) or [source:trunk/_build/latex/TheraPy.pdf](#) (pdf).

Alternatively, after downloading and unpacking the tarball or in a version obtained directly from the repository, you will have local copies of the documentation. Under the main "therapy" directory, the html version is in "_build/html" and the pdf version can be found in "_build/latex".

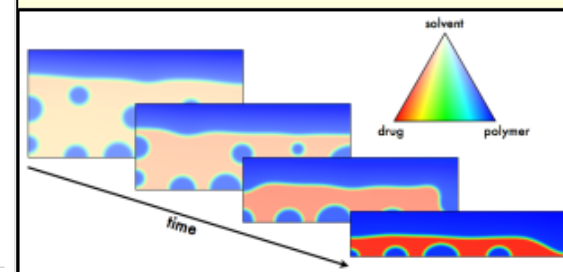
Support

The forum for open discussion of all issues related to the TheraPy scripts can be accessed by clicking the "Forums" tab at the top of the page. If you find a bug or have a specific feature request, please enter it using the "New Issue" tab.

Terms of Use

This software and documentation (the "Software") were developed at the Food and Drug Administration (FDA) by employees of the Federal Government in the course of their official duties. Pursuant to Title 17, Section 105 of the United States Code, this work is not subject to copyright protection and is in the public domain. Permission is hereby granted, free of charge, to any person obtaining a copy of the Software, to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, or sell copies of the Software or derivatives, and to permit persons to whom the Software is furnished to do so. FDA assumes no responsibility whatsoever for use by other parties of the Software, its source code, documentation or compiled executables, and makes no guarantees, expressed or implied, about its quality, reliability, or any other characteristic. Further, use of this code in no way implies endorsement by the FDA or confers any advantage in regulatory decisions. Although this software can be redistributed and/or modified freely, we ask that any derivative works bear some notice that they are derived from it, and any modified versions bear some notice that they have been modified.

Also available in: [HTML](#) | [TXT](#)



Microstructure evolution in
controlled drug delivery systems
for medical device applications

Easy Authoring and Broad Dissemination

- Matforge live CDs are
 - un-official “re-spin” of the Ubuntu 9.10 live CDs;
 - designed to help students & researchers “jump start” using a code without having to install the code and all of its dependencies.

To use the live CD,

- download and burn the ISO image into a CD;
- reboot computer from the CD.

The ISO image can also be used in a virtualization environment (like VirtualBox, VMWare, Parallels, KVM, Qemu, etc.) without needing to reboot the host computer.

How to participate?

<http://matforge.org/signup>



MatForge is a workspace for collaborative development of computational materials codes.

Please complete the following form if you are interested in utilizing MatForge to develop and communicate research and teaching resources.

If you have any questions, please [contact](#) us.

1. Project name (30 character max)

2. Short project name for URL (20 character max - lowercase, digits & dashes only)

3. Lead Project Administrator

Name

Email

Institution

Concluding Remarks

- Need to integrate computational materials research codes into undergraduate coursework
 - compelling/integral/advantageous to students
 - preparing next generation materials code developers
- Collaborative computational mat'l research & education
 - bring together significant research code and vetted undergraduate teaching resources
 - Establish consortial academe/gov't/industry materials community center with a sustainability plan
- MatForge: a collaborative partner

Contact Info

- MatForge <http://matforge.org>

Questions? Please contact:

Laura Bartolo
Kent State University
Email: lbartolo@kent.edu

MatForge is supported by the National Science Foundation (DUE-0532831, 0840792). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF.