Code Release

How I got my Jupyter Notebook published on code.usgs.gov
(with lots of help from Rich Signell, Eric Martinez, and Callie Oblinger)

Chris Sherwood
USGS Woods Hole Coastal and Marine Science Center

January, 2018

csherwood-usgs / bateman

Unwatch 1 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki

Branch: master ▾ bateman / bateman.ipynb

csherwood-usgs fit_loss in progress

1 contributor

810 lines (809 sloc) | 117 KB

Bateman Equations for Multiple Decay

The inventory of each of four compounds was calculated assuming a multi-
assumptions (Eganhouse and Pontolillo, 2008), as follows: (1) reactions a
competing reaction is loss through unspecified physical processes discussed
throughout the sediment deposit (except the physical loss rate). The time-
dechlorination to $DDMU$ and by possible losses through other proces
determined losses from dechlorination to $DDNU$, gains from dechlorina
$DDNU$ inventory changes were analogous, with losses to an unspecifie
losses through other processes. The inventory of the final, unspecified prod
and possible losses to other processes. The coupled differential equations (1

$$\frac{dC_{DDE}}{dt} = -k_1 C_{DDE} - k_4 C_{DDE}$$
$$\frac{dC_{DDMU}}{dt} = -k_2 C_{DDMU} + k_1 C_{DDE} - k_4 C_{DDMU}$$
$$\frac{dC_{DDNU}}{dt} = -k_3 C_{DDNU} + k_2 C_{DDMU} - k_4 C_{DDNU}$$
$$\frac{dC_{UN}}{dt} = k_3 C_{DDNU} - k_4 C_{UN}$$

where the molar concentrations C [μmol/kg dry sediment] of each compound and the transformation rate coefficients [y-1] are labeled with subscripts. The losses to other processes accounted for physical removal of compounds from the sediment. Processes that have been suggested include direct desorption, resuspension and desorption, desorption into porewater and irrigation, and uptake by benthic deposit feeders. We have assumed that the combined rate coefficient for these processes applies equally to all four compounds.

Import dependencies

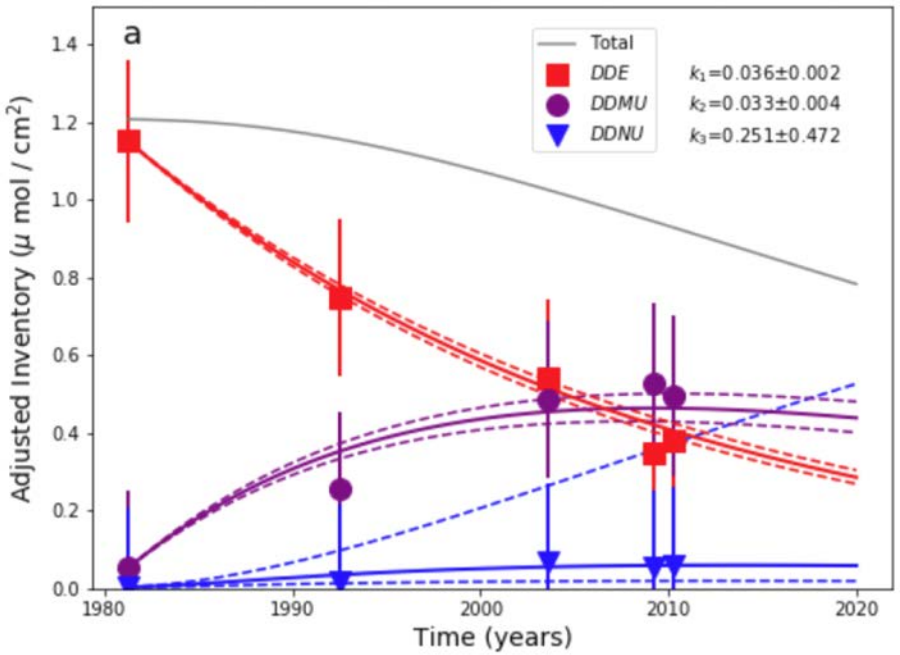
LmFit (https://lmfit.github.io/lmfit-py/) "provides a high-level interface to non-linear optimization and curve fitting problems".

In [11]:
```
from scipy.integrate import odeint
from pylab import *
from lmfit import *
```

Stand-alone example of integrating the coupled ODEs

In [14]:
```
# In my previous notebook decay rates were global variables
# so they can be seen inside the function. That is not necessary
# if you use the args = (k,) syntax
k = np.array([.046, .056, .46, .0])

# define a function to represent coupled ordinary differential eqns.
# this version allows for a fourth loss term that applies to all compounds
def dcdt(c, t, k):
    dfdt = np.zeros(4)
    dfdt[0] = c[0]* -k[0] - c[0]*k[3]
    dfdt[1] = c[1]* -k[1] + c[0]*k[0] - c[1]*k[3]
    dfdt[2] = c[2]* -k[2] + c[1]*k[1] - c[2]*k[3]
    dfdt[3] =               c[2]*k[2] - c[3]*k[3]
    return dfdt

# intial concentration for four constituents
C0e = np.array([1.34, .07, .004, 0.])

# time array
# te = linspace(0.0,100.,50)

# or, for discrete times that match Site 6C measu
te = np.array(( 1981.27, 1992.51, 2003.58, 2009.2

# Use the
Ce = odeint(dcdt,C0e,te, args = (k,))
```

Figure 7



Summary

Scratch Code on Computer or GitHub

- No formal approval required
- Good to have appropriate license & disclaimer
- No PII / passwords / sensitive information

Preliminary Source Code

- Requires science center director approval for preliminary release.
- Requires an administrative security review for all contributions.
- Requires appropriate license, disclaimer (preliminary software), and metadata.

Approved Source Code

- Requires science center director approval for approved release.
- Requires administrative security (one), code (one), and subject matter expert (two) reviews for all contributions.
- Requires appropriate license, disclaimer (approved software), and metadata.

Administrative / Security Review

Any trusted person; entire revision history

- Personally identifiable information (PII)
- Absolute file system paths
- Internal server host names or IP addresses
- Usernames/Passwords

Code Review

Trusted developer with applicable programming knowledge

- Coding standards
- Unit tests passing
- User input cleansing
- Memory leaks
- Vulnerabilities
- Optimizations

Subject Matter Expert Review

Trusted scientific expert with applicable subject-matter knowledge as well as sufficient applicable programming knowledge

- Scientific reviews ensure the software generates output that align with published or otherwise well-known expected results. This may involve:
- Comparing output with external data sets
- Comparing algorithms with published scholarly articles about the algorithm
- Reviewing unit and integration test results

IPDS Flow

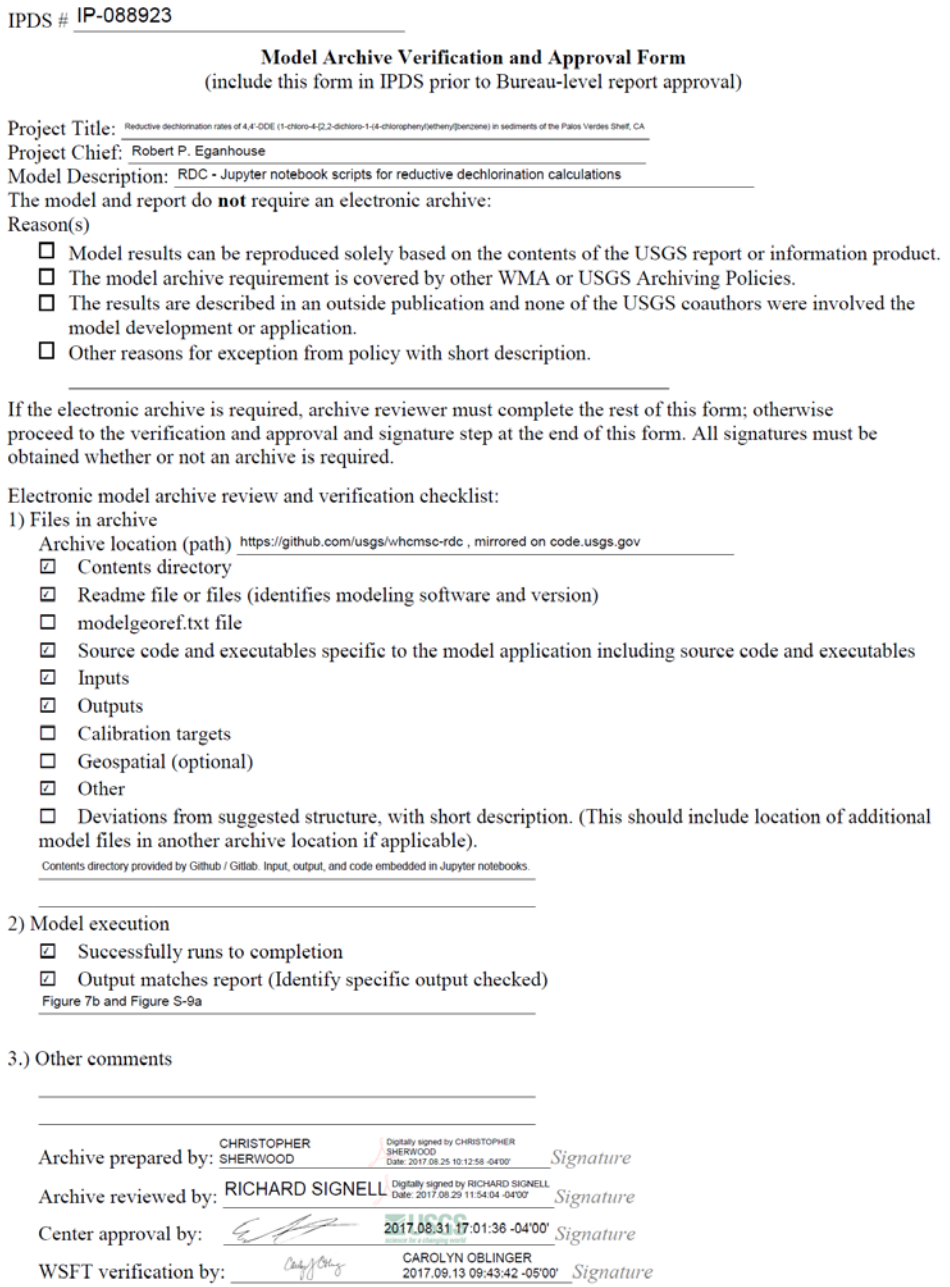
This process was specific to Water, and followed Office of Groundwater Tech. Mem. 2015.2.

The IPDS number was associated with the journal article.

Signatures were required from author, lead reviewer, Center Director, and one of the three WSFT specialists.

I minted the doi using the USGS DOI Creation Tool https://www1.usgs.gov/csas/doi/

At our Center, we don't have a form, but can release code its own IPDS number with same review process.



IPDS # IP-088923

Model Archive Verification and Approval Form
(include this form in IPDS prior to Bureau-level report approval)

Project Title: Reductive dechlorination rates of 4,4'-DDE (1-chloro-4-[2,2-dichloro-1-(4-chlorophenyl)ethenyl]benzene) in sediments of the Palos Verdes Shelf, CA
Project Chief: Robert P. Eganhouse
Model Description: RDC - Jupyter notebook scripts for reductive dechlorination calculations
The model and report do **not** require an electronic archive:
Reason(s)

- ☐ Model results can be reproduced solely based on the contents of the USGS report or information product.
- ☐ The model archive requirement is covered by other WMA or USGS Archiving Policies.
- ☐ The results are described in an outside publication and none of the USGS coauthors were involved the model development or application.
- ☐ Other reasons for exception from policy with short description.

If the electronic archive is required, archive reviewer must complete the rest of this form; otherwise proceed to the verification and approval and signature step at the end of this form. All signatures must be obtained whether or not an archive is required.

Electronic model archive review and verification checklist:
1) Files in archive
Archive location (path) https://github.com/usgs/whcmsc-rdc , mirrored on code.usgs.gov

- ☑ Contents directory
- ☑ Readme file or files (identifies modeling software and version)
- ☐ modelgeoref.txt file
- ☑ Source code and executables specific to the model application including source code and executables
- ☑ Inputs
- ☑ Outputs
- ☐ Calibration targets
- ☐ Geospatial (optional)
- ☑ Other
- ☐ Deviations from suggested structure, with short description. (This should include location of additional model files in another archive location if applicable).

Contents directory provided by Github / Gitlab. Input, output, and code embedded in Jupyter notebooks.

2) Model execution

- ☑ Successfully runs to completion
- ☑ Output matches report (Identify specific output checked)

Figure 7b and Figure S-9a

3.) Other comments

Archive prepared by: CHRISTOPHER SHERWOOD Digitally signed by CHRISTOPHER SHERWOOD Date: 2017.08.25 10:12:58 -04'00' *Signature*
Archive reviewed by: RICHARD SIGNELL Digitally signed by RICHARD SIGNELL Date: 2017.08.29 11:54:04 -04'00' *Signature*
Center approval by: 2017.08.31 17:01:36 -04'00' *Signature*
WSFT verification by: CAROLYN OBLINGER 2017.09.13 09:43:42 -05'00' *Signature*

Code Flow

https://github.com/csherwood-usgs/bateman

Working repo with lots of junk and eventually a final version

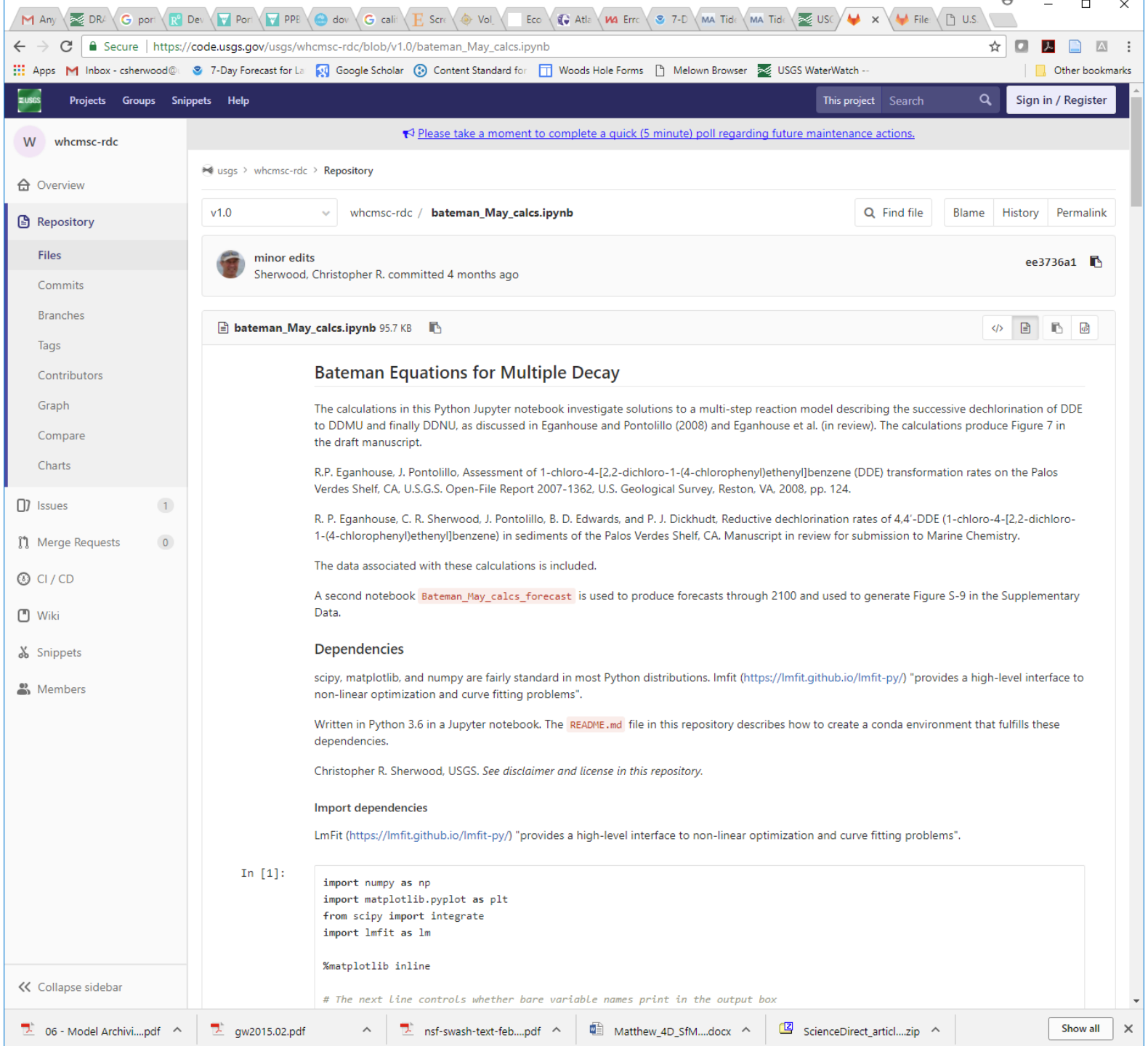
https://github.com/csherwood-usgs/whcmsc-rdc

Cleaned up code, licenses, disclaimers, and environment.yml with minimum requirements. This was the version reviewed. After review, made a `release` branch, replaced the Preliminary disclaimer with the Approved disclaimer. Then made a tagged `v1.0` for release. Eric mirrored this to:

Code.usgs.gov is GitLab, so some of the markdown (esp. math) is different. I had to go the lowest-common denominator)

https://code.usgs.gov/usgs/whcmsc-rdc/tree/v1.0

Ta Da!


https://code.usgs.gov/usgs/whcmsc-rdc/blob/v1.0/bateman_May_calcs.ipynb
Projects Groups Snippets Help
Sign in / Register
whcmsc-rdc
Please take a moment to complete a quick (5 minute) poll regarding future maintenance actions.
usgs > whcmsc-rdc > Repository
Overview
Repository
Files
Commits
Branches
Tags
Contributors
Graph
Compare
Charts
Issues
Merge Requests
CI / CD
Wiki
Snippets
Members
Collapse sidebar
v1.0
whcmsc-rdc / bateman_May_calcs.ipynb
Find file
Blame
History
Permalink
minor edits
Sherwood, Christopher R. committed 4 months ago
ee3736a1
bateman_May_calcs.ipynb 95.7 KB
Bateman Equations for Multiple Decay
The calculations in this Python Jupyter notebook investigate solutions to a multi-step reaction model describing the successive dechlorination of DDE to DDMU and finally DDNU, as discussed in Eganhouse and Pontolillo (2008) and Eganhouse et al. (in review). The calculations produce Figure 7 in the draft manuscript.
R.P. Eganhouse, J. Pontolillo, Assessment of 1-chloro-4-[2,2-dichloro-1-(4-chlorophenyl)ethenyl]benzene (DDE) transformation rates on the Palos Verdes Shelf, CA, U.S.G.S. Open-File Report 2007-1362, U.S. Geological Survey, Reston, VA, 2008, pp. 124.
R. P. Eganhouse, C. R. Sherwood, J. Pontolillo, B. D. Edwards, and P. J. Dickhudt, Reductive dechlorination rates of 4,4'-DDE (1-chloro-4-[2,2-dichloro-1-(4-chlorophenyl)ethenyl]benzene) in sediments of the Palos Verdes Shelf, CA. Manuscript in review for submission to Marine Chemistry.
The data associated with these calculations is included.
A second notebook Bateman_May_calcs_forecast is used to produce forecasts through 2100 and used to generate Figure S-9 in the Supplementary Data.
Dependencies
scipy, matplotlib, and numpy are fairly standard in most Python distributions. lmfit (https://lmfit.github.io/lmfit-py/) "provides a high-level interface to non-linear optimization and curve fitting problems".
Written in Python 3.6 in a Jupyter notebook. The README.md file in this repository describes how to create a conda environment that fulfills these dependencies.
Christopher R. Sherwood, USGS. See disclaimer and license in this repository.
Import dependencies
LmFit (https://lmfit.github.io/lmfit-py/) "provides a high-level interface to non-linear optimization and curve fitting problems".
In [1]:
import numpy as np
import matplotlib.pyplot as plt
from scipy import integrate
import lmfit as lm
%matplotlib inline
The next line controls whether bare variable names print in the output box
06 - Model Archivi....pdf
gw2015.02.pdf
nsf-swash-text-feb....pdf
Matthew_4D_SfM....docx
ScienceDirect_articl....zip
Show all

Links

- https://www2.usgs.gov/usgs-manual/im/IM-OSQI-2016-01.html
- https://github.com/usgs/best-practices/blob/master/software/reviews.md#administrative-security-review
- https://www1.usgs.gov/csas/doi/
- https://water.usgs.gov/ogw/policy/gw-model/index.html

Core Science Analytics, Synthesis, and Libraries (CSAS&L) - Digital Object Identifier (DOI) Service

Create DOI

⚠ Do **not** create fake or 'test' records in the DOI Tool. Create DOIs only for **actual** data products you intend to release.

MyDOIs | Search DOIs | USGS DOI Stats | DOI Stats by Data Source

Edit My DOI:

Show 50 entries

Search:

DOI	Title	Status	DOI URL	Last Modified	Authors	Primary Users	Backup Users	Link Checking Status
doi:10.5066/F7ZK5FMH	RDC - Repository with Python Jupyter Notebooks for reductive dechlorination calculations	Published	Copy URL	2017-11-27	Sherwood, Christopher R.	csherwood@usgs.gov		Valid Link

Showing 1 to 1 of 1 entries

Previous 1 Next