MITgcm git info etc…

Overview
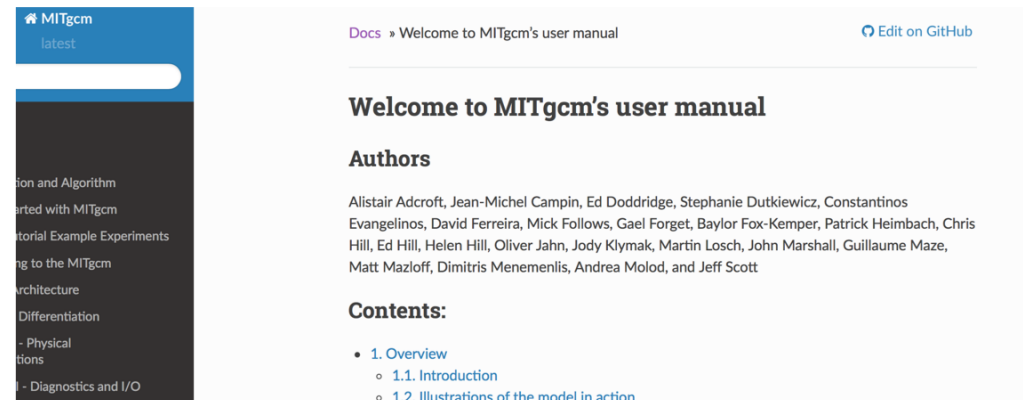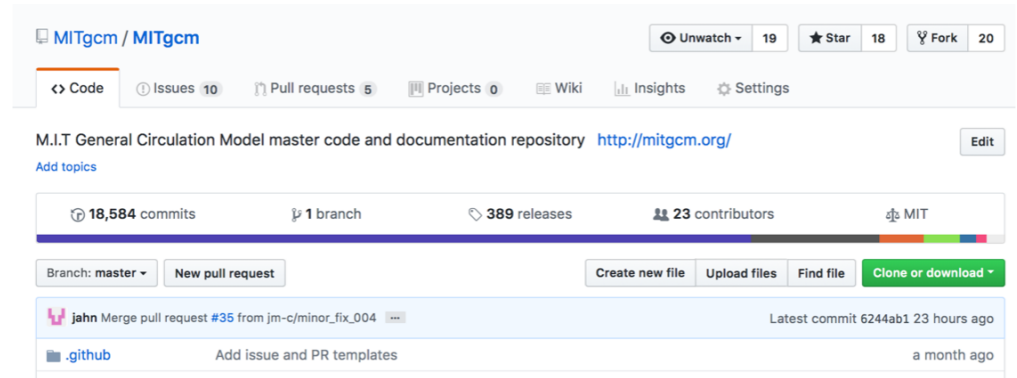
- Code distribution and development now hosted under git using GitHub as main repository
  - https://github.com/MITgcm/MITgcm

- Documentation manual within the main repository using .rst format and published through readthedocs.
  - http://mitgcm.readthedocs.io

http://mitgcm.org updated to point to new spaces (old links also there for reference).

Topics for today

- A little about about git setup
  - New ways to
    - Download (git clone)
    - Browse code (LXR)
    - Provide code/doc feedback, requests and contributions (issues, PR)

- Documentation update status, plans and feedback discussion
  - Current status
  - New sections
  - Areas for contribution

- http://mitgcm.org web skin upgrade previewing soon

http://mitgcm.readthedocs.io/en/latest/getting_started/getting_started.html#method-1

```
% git clone https://github.com/MITgcm/MITgcm.git
```

http://lxr.mitgcm.org/lxr2/source

MITgcm

MITgcm/

1 / **MITgcm**

ⓘ Issues 10   ⑂ Pull requests 5

https://github.com/MITgcm/MITgcm/issues

https://github.com/MITgcm/MITgcm/pulls

Downloading from git

Step by step in

http://mitgcm.readthedocs.io/en/latest/getting_started/getting_started.html

3.2.1. Method 1

This section describes how to download git-aware copies of the repository. In a terminal window, cd to the directory where you want your code to reside. Type:

```
% git clone https://github.com/MITgcm/MITgcm.git
```

This will download the latest available code. If you now want to revert this code to a specific checkpoint release, first `cd` into the MITgcm directory you just downloaded, then type `git checkout checkpointXXX` where `XXX` is the checkpoint version.

Alternatively, if you prefer to use ssh keys (say for example, you have a firewall which won't allow https download), type:

```
% git clone git@github.com:MITgcm/MITgcm.git
```

You will need a GitHub account for this, and will have to generate a ssh key though your GitHub account user settings.

3.2.2. Method 2

This section describes how to do a one-time download of the MITgcm, NOT git-aware. In a terminal window, `cd` to the directory where you want your code to reside. To obtain the current code, type:

```
% wget https://github.com/MITgcm/MITgcm/archive/master.zip
```

For specific checkpoint release `XXX`, instead type:

```
% wget https://github.com/MITgcm/MITgcm/archive/checkpointXXX.zip
```

Searching code tree (LXR)

Direct access
http://lxr.mitgcm.org

Integrated into doc e.g.

In general, the horizontal momentum time-stepping can contain some terms that are treated implicitly in time, such as the vertical viscosity when using the backward time-stepping scheme (implicitViscosity =.TRUE.). The method used to solve those implicit terms is provided in Section 2.6, and equations (2.2) and (2.3) to give:

$$u^{n+1} - \Delta t \partial_z A_v \partial_z u^{n+1} + \Delta t g \partial_x \eta^{n+1} = u^n + \Delta t G_u^{(n+1/2)}$$
$$v^{n+1} - \Delta t \partial_z A_v \partial_z v^{n+1} + \Delta t g \partial_y \eta^{n+1} = v^n + \Delta t G_v^{(n+1/2)}$$

http://mitgcm.readthedocs.io/en/latest/algorithm/algorithm.html

Definitions for implicitViscosity

Type	Member of	File	Line
local and common block variable		/model/inc/PARAMS.h	✳437
global or module variable		/model/inc/PARAMS.h	✳528
local and common block variable		/model/src/ini_parms.F	✳213

3 declarations in 2 files.

References to implicitViscosity

File	Line
/model/inc/PARAMS.h	✳437 ✳528
/model/src/config_check.F	✳801 ✳811
/model/src/config_summary.F	✳430
/model/src/dynamics.F	✳578 ✳586 ✳588 ✳621
/model/src/ini_parms.F	✳213
/model/src/set_defaults.F	✳204
/model/src/set_parms.F	✳66
/pkg/ggl90/ggl90_check.F	✳94
/pkg/kl10/kl10_check.F	✳62
/pkg/kpp/kpp_check.F	✳141
/pkg/mom_common/mom_u_implicit_r.F	✳84 ✳280
/pkg/mom_common/mom_v_implicit_r.F	✳84 ✳280
/pkg/mom_fluxform/mom_fluxform.F	✳239 ✳524 ✳553 ✳756 ✳785
/pkg/mom_vecinv/mom_vecinv.F	✳444 ✳517
/pkg/my82/my82_check.F	✳67
/pkg/pp81/pp81_check.F	✳60

http://lxr.mitgcm.org/lxr2/ident/MITgcm?_i=implicitViscosity

Feedback and contributions to code/doc.

Issues provide feedback, bug reports related to code or documentation
- Not for support questions
- Some discussion, but targeted/focused to an artefact that can ultimately be resolved and closed (i.e. not open ended debate).

Pull requests (PRs) allow anybody to submit new or modified code. Submissions will be reviewed etc… and, if useful and adequate quality, accepted. PRs can be used for code, test cases, documentation etc…

Current issue list

https://github.com/MITgcm/MITgcm/issues

Current pull requests
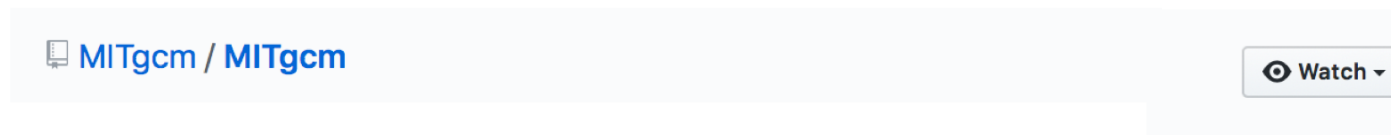

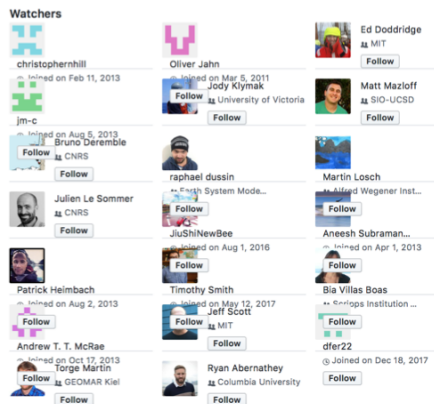https://github.com/MITgcm/MITgcm/pulls

More detailed instructions.

http://mitgcm.readthedocs.io/en/latest/contributing/contributing.html#contributing-to-the-mitgcm

Tracking and watching
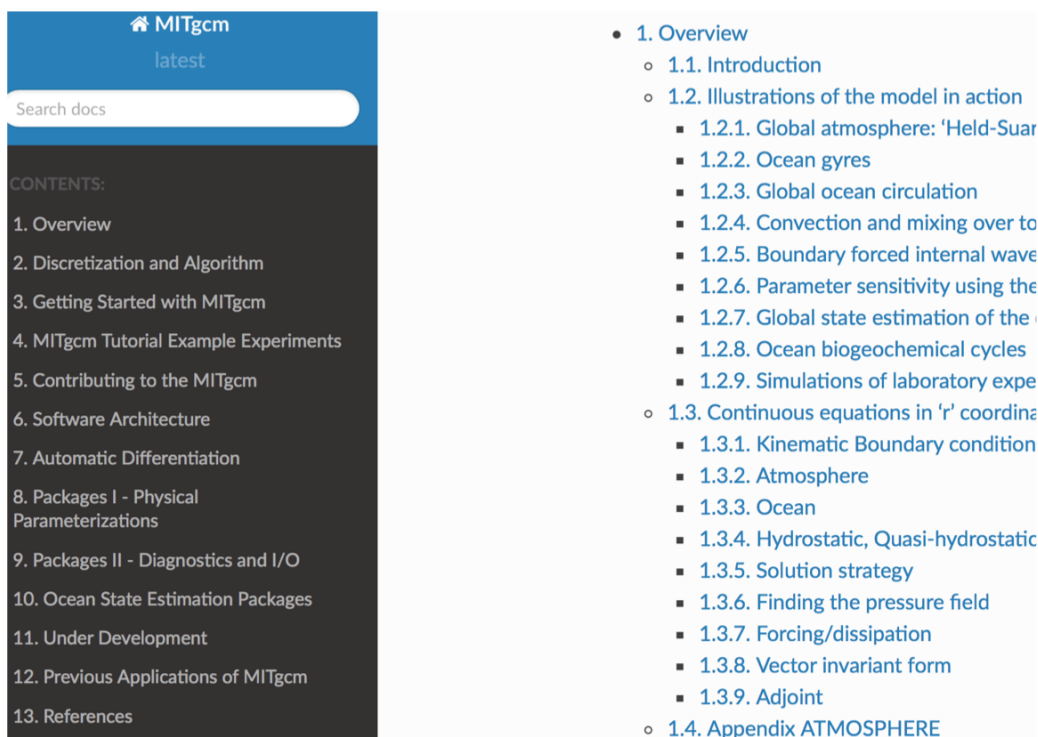
- Good way to keep track of new items

- Currently have only 20 watchers, feel free to join!

Documentation

http://mitgcm.readthedocs.io/en/latest/

Nearly all of existing documentation has been converted from Latex to .rst and migrated from CVS to git.

One remaining major piece TBD
- *Packages II: Diagnostics and I/O*

Previous and current/future manual "versions" being integrated into Dspace (and Zenodo?) to provide better citable docs.
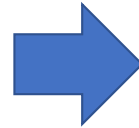
Planning to "overhaul" tutorials, getting started info. So only two tutorials currently migrated.

Latex to .rst mixes automatic (pandoc) and manual ( ).

The horizontal momentum and continuity equations for th
(\ref{eq:ocean-mom} and \ref{eq:ocean-cont}), or for th
(\ref{eq:atmos-mom} and \ref{eq:atmos-cont}), can be su
\begin{eqnarray}
\partial_t u + g \partial_x \eta & = & G_u \\
\partial_t v + g \partial_y \eta & = & G_v \\
\partial_x u + \partial_y v + \partial_z w & = & 0
\end{eqnarray}
where we are adopting the oceanic notation for brevity.
the momentum equations, except for surface pressure gra
encapsulated in the G vector. The continuity equation
integrated over the fluid depth, H, and with the rigi
flow boundary conditions applied, becomes:

.. _press_meth_rigid:

Pressure method with rigid-lid
==============================

The horizontal momentum and continuity equations for the ocean
(:eq:`eq-ocean-mom` and :eq:`eq-ocean-cont`), or for the atmos
(:eq:`atmos-mom` and :eq:`atmos-cont`), can be summarized by:

.. math::

    \begin{aligned}
    \partial_t u + g \partial_x \eta & = & G_u \\
    \partial_t v + g \partial_y \eta & = & G_v \\
    \partial_x u + \partial_y v + \partial_z w & = & 0\end{align

where we are adopting the oceanic notation for brevity. All te
the momentum equations, except for surface pressure gradient, a
encapsulated in the :math:`G` vector. The continuity equation,
integrated over the fluid depth, :math:`H`, and with the rigid-
normal flow boundary conditions applied, becomes:

https://github.com/MITgcm/MITgcm/tree/master/doc

Tutorial/getting started overhaul

Once latex to .rst conversion is fully complete one focus will be on overhaul of getting started/tutorial sections including examining
 - basic getting started
 - using more features
 - research relevant starting points
 - regression/verification suite

Starting with new realistic lat-lon setup suite at 2.8 degree resolution. Current scope
- Basic physical setup (z-coord)
- Basic physical setup (p-coord)
- Carbon cycle model
- Simple adjoint sensitivity

See/comment at https://github.com/MITgcm/MITgcm/issues/45

Related projects section

- Replacing "previous applications".

- A first cut is at
  https://github.com/MITgcm/MITgcm/pull/47/files

- Send PRs with things to add (once initial examples are merged).

- Any near-term comments can be added to
  https://github.com/MITgcm/MITgcm/pull/47

```
 5  +Projects Related to MITgcm
 6  +=========================
 7  +
 8  +Estimating the Circulation and Climate of the Ocean (ECCO)
 9  +----------------------------------------------------------
10  +
11  +ECCO is an oceanic reanalysis product that makes use of MITgcm's adjoint capabilities to assimilate observations into a
     dynamically self-consistent state estimate.
12  +
13  +website: https://ecco.jpl.nasa.gov/
14  +
15  +
16  +Octopus - Lagrangian Particle Tracker
17  +-------------------------------------
18  +
19  +Octopus is an offline Lagrangian particle tracker that uses saved velocity fields from MITgcm simulations.
20  +
21  +website: https://github.com/jinbow/Octopus
22  +
23  +
24  +Southern Ocean State Estimation (SOSE)
25  +--------------------------------------
26  +
27  +SOSE uses the same techniques as ECCO to produce an eddy-permitting state estimate of the Southern Ocean.
28  +
29  +website: http://sose.ucsd.edu/
30  +
31  +
32  +xmitgcm
33  +-------
34  +
35  +xmitgcm is a Python module that loads MITgcm MDS output files as `xarray <http://xarray.pydata.org/en/stable/>`_ arrays
     with the associated grid information. These can be easily exported as NetCDF files.
36  +
37  +website: http://xmitgcm.readthedocs.io/en/latest/
```

Documentation (and code) contributions/feedback.

- Looking for feedback on documentation and plans (e.g. https://github.com/MITgcm/MITgcm/issues/45).

- Comments/feedback on all/any areas very welcome (not just getting started/tutorials)

- Other example experiment contributions always welcome (can we make this easier?).

- Package documentation is always welcome. Ryan is working on pkg/layers for example.

- Planning to incorporate more code and documentation for self-organizing ecosystem work (GUD).

New web "skin" preview coming soon

Updating web site to integrate with new doc, code.

New "skin" to freshen up.

Monthly stories and publication summaries ongoing etc....