MITgcm code and documentation repo updates

- Jean-Michel Campin, Jeff Scott, Ed Doddridge, Oliver Jahn, Chris Hill

  - Main activities

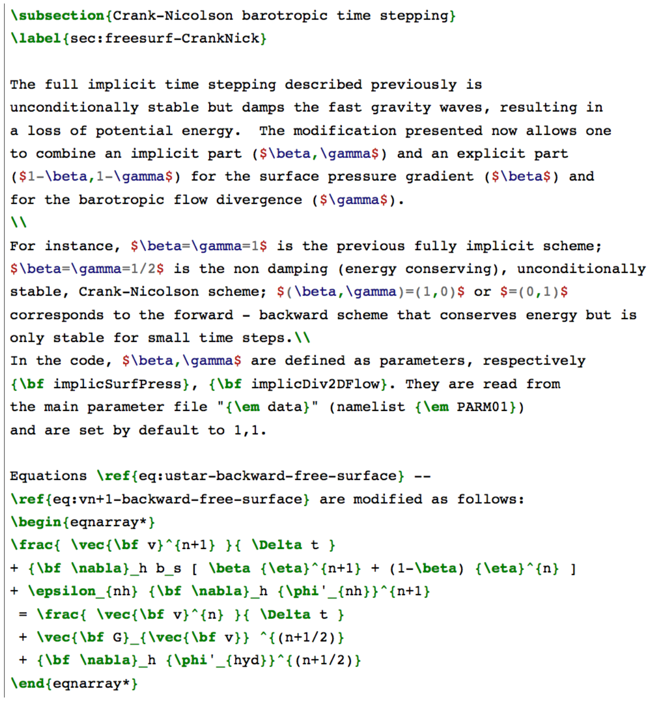
  1. Migrating from own server CVS to hosted git (at github.com)
  2. Migrating from own server .tex for documentation to hosted “github, .rst + sphinx/readthedocs pipeline”
  3. Migrating code browser to LXR
  4. Web site new skin

MITgcm CVS to git

- Test setup is at
  - https://github.com/altMITgcm/MITgcm
  - Mostly follows pre-existing directory structure except documentation is now within the same directory tree (and in a different format – see later)
  - Currently testing pull request (PR) workflows for both code and documentation
    - 4 people with github PR accept auth (Jean-Michel, Oliver, Chris, Ed)
    - 2 approvals required to merge
    - Travis rudimentary automated testing
    - HowTo guide here http://mitgcm-example.readthedocs.io/en/latest/contributing/contributing.html
  - Note - path names (github.com/altMITgcm/MITgcm and mitgcm-example.readthedocs.io) are intermediate. Things will ultimately migrate to github.com/MITgcm/MITgcm and mitgcm.readthedocs.io

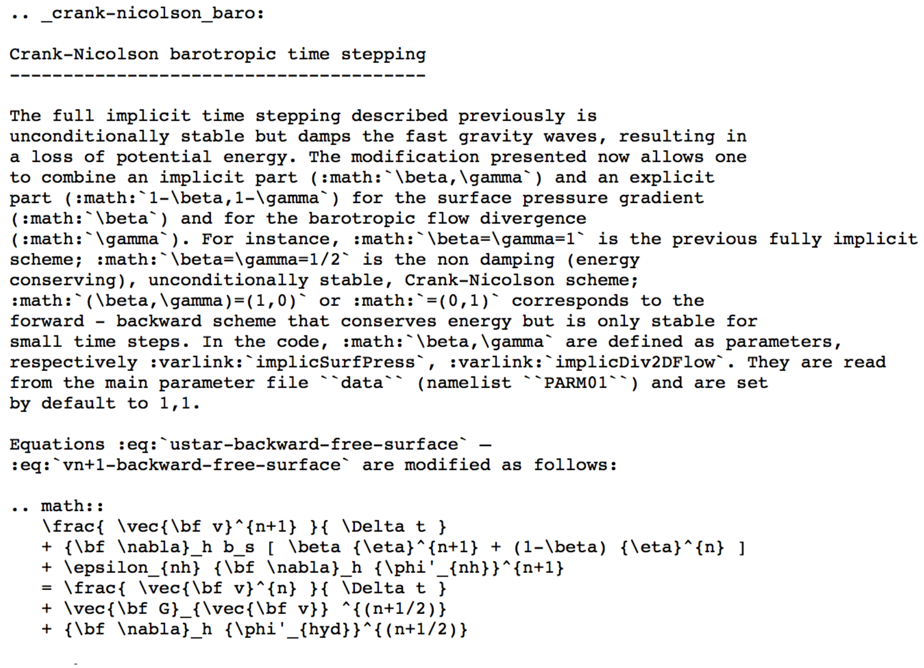
MITgcm latex to rst

- Jeff Scott and Ed Doddridge have been converting several parts of manual to .rst
  - Current state is at http://mitgcm-example.readthedocs.io/en
  - Some conversion is automated using pandoc, but some is by hand
  - .rst uses mostly same math syntax as latex – but not identical e.g.



```
\subsection{Crank-Nicolson barotropic time stepping}
\label{sec:freesurf-CrankNick}

The full implicit time stepping described previously is
unconditionally stable but damps the fast gravity waves, resulting in
a loss of potential energy.  The modification presented now allows one
to combine an implicit part ($\beta,\gamma$) and an explicit part
($1-\beta,1-\gamma$) for the surface pressure gradient ($\beta$) and
for the barotropic flow divergence ($\gamma$).
\\
For instance, $\beta=\gamma=1$ is the previous fully implicit scheme;
$\beta=\gamma=1/2$ is the non damping (energy conserving), unconditionally
stable, Crank-Nicolson scheme; $(\beta,\gamma)=(1,0)$ or $=(0,1)$
corresponds to the forward - backward scheme that conserves energy but is
only stable for small time steps.\\
In the code, $\beta,\gamma$ are defined as parameters, respectively
{\bf implicSurfPress}, {\bf implicDiv2DFlow}. They are read from
the main parameter file "{\em data}" (namelist {\em PARM01})
and are set by default to 1,1.

Equations \ref{eq:ustar-backward-free-surface} --
\ref{eq:vn+1-backward-free-surface} are modified as follows:
\begin{eqnarray*}
\frac{ \vec{\bf v}^{n+1} }{ \Delta t }
+ {\bf \nabla}_h b_s [ \beta {\eta}^{n+1} + (1-\beta) {\eta}^{n} ]
+ \epsilon_{nh} {\bf \nabla}_h {\phi'_{nh}}^{n+1}
 = \frac{ \vec{\bf v}^{n} }{ \Delta t }
 + \vec{\bf G}_{\vec{\bf v}} ^{(n+1/2)}
 + {\bf \nabla}_h {\phi'_{hyd}}^{(n+1/2)}
\end{eqnarray*}
```

http://wwwcvs.mitgcm.org/viewvc/MITgcm/manual/s_algorithm/text/time_stepping.tex?revision=1.34&view=markup

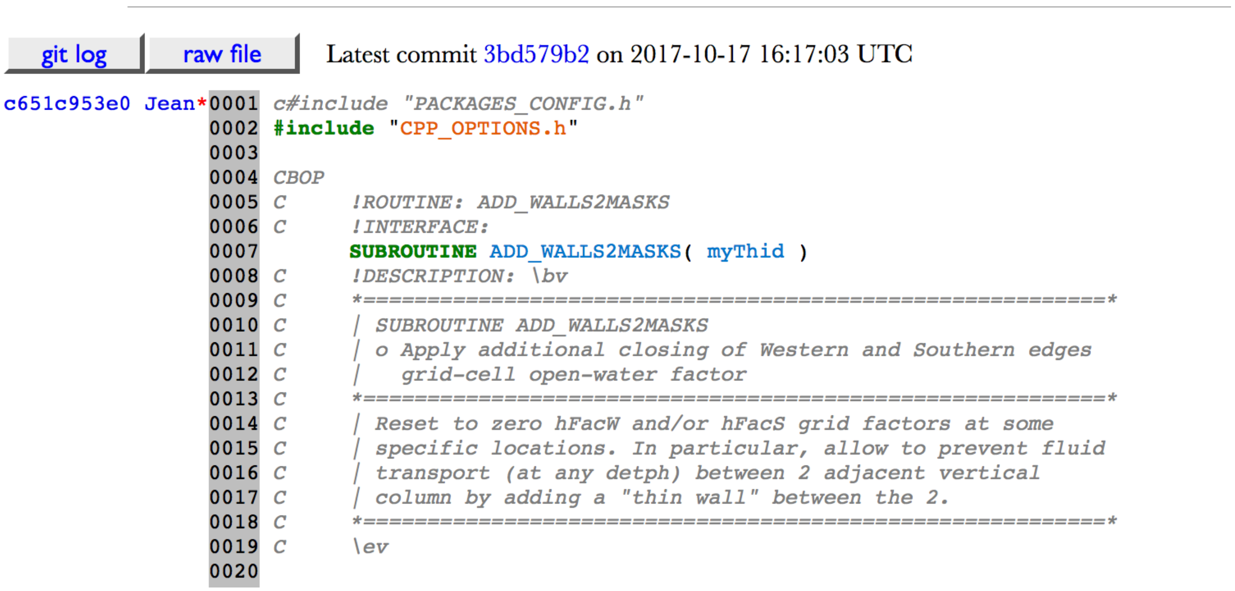


```
.. _crank-nicolson_baro:

Crank-Nicolson barotropic time stepping
---------------------------------------

The full implicit time stepping described previously is
unconditionally stable but damps the fast gravity waves, resulting in
a loss of potential energy. The modification presented now allows one
to combine an implicit part (:math:`\beta,\gamma`) and an explicit
part (:math:`1-\beta,1-\gamma`) for the surface pressure gradient
(:math:`\beta`) and for the barotropic flow divergence
(:math:`\gamma`). For instance, :math:`\beta=\gamma=1` is the previous fully implicit
scheme; :math:`\beta=\gamma=1/2` is the non damping (energy
conserving), unconditionally stable, Crank-Nicolson scheme;
:math:`(\beta,\gamma)=(1,0)` or :math:`=(0,1)` corresponds to the
forward - backward scheme that conserves energy but is only stable for
small time steps. In the code, :math:`\beta,\gamma` are defined as parameters,
respectively :varlink:`implicSurfPress`, :varlink:`implicDiv2DFlow`. They are read
from the main parameter file ``data`` (namelist ``PARM01``) and are set
by default to 1,1.

Equations :eq:`ustar-backward-free-surface` –
:eq:`vn+1-backward-free-surface` are modified as follows:

.. math::
   \frac{ \vec{\bf v}^{n+1} }{ \Delta t }
   + {\bf \nabla}_h b_s [ \beta {\eta}^{n+1} + (1-\beta) {\eta}^{n} ]
   + \epsilon_{nh} {\bf \nabla}_h {\phi'_{nh}}^{n+1}
   = \frac{ \vec{\bf v}^{n} }{ \Delta t }
   + \vec{\bf G}_{\vec{\bf v}} ^{(n+1/2)}
   + {\bf \nabla}_h {\phi'_{hyd}}^{(n+1/2)}
```

https://raw.githubusercontent.com/altMITgcm/MITgcm/master/doc/discret_algorithm/crank-nicol.rst

Code browser to lxr (with a few fortran customizations!)

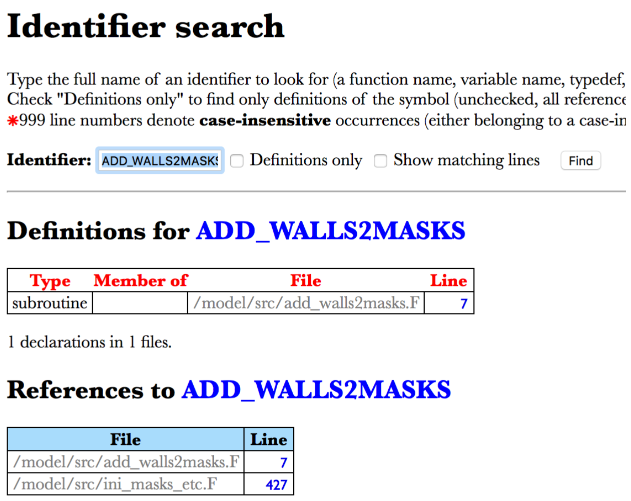
- e.g. http://lxr.mitgcm.org/lxr2/source/MITgcm/model/src/add_walls2masks.F



Latest commit 3bd579b2 on 2017-10-17 16:17:03 UTC

c651c953e0 Jean*

```
c#include "PACKAGES_CONFIG.h"
#include "CPP_OPTIONS.h"

CBOP
C     !ROUTINE: ADD_WALLS2MASKS
C     !INTERFACE:
      SUBROUTINE ADD_WALLS2MASKS( myThid )
C     !DESCRIPTION: \bv
C     *==========================================================*
C     | SUBROUTINE ADD_WALLS2MASKS
C     | o Apply additional closing of Western and Southern edges
C     |   grid-cell open-water factor
C     *==========================================================*
C     | Reset to zero hFacW and/or hFacS grid factors at some
C     | specific locations. In particular, allow to prevent fluid
C     | transport (at any detph) between 2 adjacent vertical
C     | column by adding a "thin wall" between the 2.
C     *==========================================================*
C     \ev

```



Identifier search

Type the full name of an identifier to look for (a function name, variable name, typedef,
Check "Definitions only" to find only definitions of the symbol (unchecked, all reference
*999 line numbers denote **case-insensitive** occurrences (either belonging to a case-in

Identifier: ADD_WALLS2MASKS ☐ Definitions only ☐ Show matching lines

Definitions for ADD_WALLS2MASKS

Type	Member of	File	Line
subroutine		/model/src/add_walls2masks.F	7

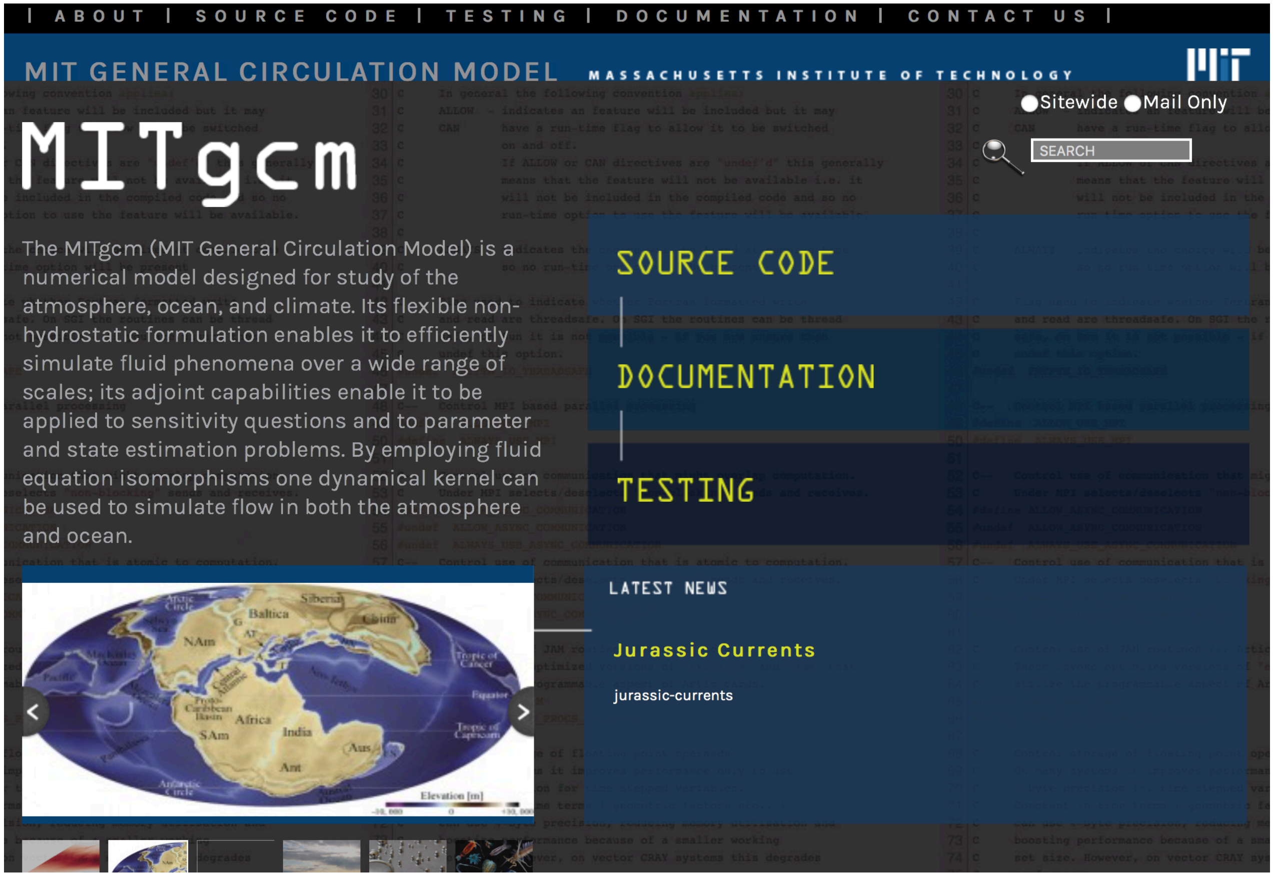
1 declarations in 1 files.

References to ADD_WALLS2MASKS

File	Line
/model/src/add_walls2masks.F	7
/model/src/ini_masks_etc.F	427

Provides most of same name look up and occurrence capabilities.
Temporarily hosted on Raspberry Pi (?) so a bit slow.

Web site new skin



| ABOUT | SOURCE CODE | TESTING | DOCUMENTATION | CONTACT US |
MIT GENERAL CIRCULATION MODEL
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
○Sitewide ○Mail Only
SEARCH

MITgcm

The MITgcm (MIT General Circulation Model) is a numerical model designed for study of the atmosphere, ocean, and climate. Its flexible non-hydrostatic formulation enables it to efficiently simulate fluid phenomena over a wide range of scales; its adjoint capabilities enable it to be applied to sensitivity questions and to parameter and state estimation problems. By employing fluid equation isomorphisms one dynamical kernel can be used to simulate flow in both the atmosphere and ocean.

LATEST NEWS

Jurassic Currents

jurassic-currents

~Timeline

- Feedback and suggestions welcome at all times!

- CVS MITgcm/ and manual/ freeze and production git switch
  - Hopefully in next few weeks (pendign feedback/suggestions)
  - New .rst documentation will not be complete, but is a long way along. Looking for contributions

- MITgcm_contrib/ likely staying under CVS for now

- LXR will migrate to faster server and storage to hopefully speed up queries like http://lxr.mitgcm.org/lxr2/ident/MITgcm?_i=myThid

- New web site skin demos shortly after MITgcm/ and manual/ to final github locations.