TiCOM Documentation

Version 5
Last updated: February 26, 2022

Contents

1 Filestructure **2**

1.1 TiRADE & Output 2
1.2 Citcom & Output 3
1.3 Generated Plots 4

2 Usage at the Command-Line **5**

2.1 Default Settings/The 'settings' File 7
2.2 Diagnostic Mode 10
2.3 Re-making TiRADE/Citcom Source Code 13
2.4 Directories & Pathname Overloading 13
2.5 Step Overloading 14
2.6 Additional Notes 14

3 Miscellaneous **15**

3.1 Bugs 15
3.2 Future Work 15

4 Contact **16**

1 Filestructure

The following are required for TiCOM to run:

- TiRADE source code
- Citcom source code
- TiCOM main script (“`TiCOM_bash`”)
- Settings file (“`settings`”) - Defines some of the parameters that TiCOM uses
- Functions file (“`functions`”) - Contains critical functions & utility functions for TiCOM. Can be sourced & used without TiCOM

1.1 TiRADE & Output

TiRADE’s core source code consists of the following files/directories:

- `/src/` - Contains TiRADE source code files, written in C.
  - `tirade.c` - Computes the conductive temperature structure, and corresponding tidal heating distribution for a spherically symmetric planet or moon composed of an arbitrary number of layers.

    The user provides an input file containing the mechanical properties for each layer in the body, its orbital and global properties, and information about which output is needed. The program computes the tidal heating distribution using the `tidal_heating` code, then computes the conductive temperature structure based on that heating, adjusting the shell thickness in order to get an equilibrium solution, using the conduction code. The viscosity of each layer is updated based on the temperature, and the tidal heating is recomputed based on the new viscosity and shell thickness. The temperature is computed from the new heating. The program iterates until ...

    (a) ... it converges to a solution.
    (b) ... the shell melts.
    (c) ... the ocean freezes.
  - `tirade.h` - Contains macros & struct definitions necessary for TiRADE.
  - `utility.c` - Contains data transfer & various file IO helper functions that can be used to better user-friendliness.
  - `tidal_module.c` - Routines to compute the spatially-varying tidal heating within a spherically symmetric planet or moon composed of an arbitrary number of layers. Originally intended as an input parameter for a convection program such as Citcom (i.e. spatially varying internal heating, but the average properties can be analyzed on their own).

  - conduction_module.c - Routines to compute the conductive temperature structure for a spherically symmetric planet or moon composed of an arbitrary number of layers. Originally designed to see if the heat flux generated by tidal dissipation in an ice shell is consistent with a conductive temperature profile. Because the tidal heating is a function of the temperature-dependent ice viscosity, and the shell thickness, the heating should be solved at every iteration as well. This code should be coupled with tidal_heating code in order to do this.

- /bin/ - Contains TiRADE binaries.

  - tirade.x - TiRADE executable. Only exists after make tirade.x rule is used. Human-unreadable.

- /obj/ - Stores object files for corresponding /src/ files.

  - conduction_module.o
  - tidal_module.o
  - tirade.o
  - utility.o

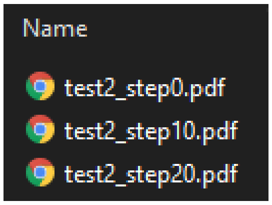
- Makefile - Compiles & links TiRADE source code. Used for make and make clean rules.

The location to which TiRADE data is outputted is defined by the TiRADE input file. If TiRADE is run independently, the output location is defined in the sole input file; if TiCOM is run, the output location is defined in the TiCOM input script, and this can be overloaded using the 'data=...' flag. In the TiRADE data subdirectory, we have a directory /input_files/ that has all of the generated TiRADE input files & we also have /step{number}/ directories for the generated TiRADE output per each step.

1.2 Citcom & Output

All of Citcom's many source files should be in the Citcom source code directory. These consist of object files, and C/C header files. TiCOM also expects an initial Citcom input file in the source code directory, which can be overloaded at the command line. In this directory, the citcom.x executable is generated after running make citcom.x.

Citcom also has its data output to the same parent directory as TiRADE's output data; the current run data is stored in the directory named after the CASE variable (default test2), which also can be overloaded at the command line. *Another* subdirectory inside that one called /post/, containing the data that the postprocessing scripts need to properly run. We also have a directory named /input_files/, containing the Citcom input files made using the make Citcom input scripts. If the --o flag is used to organize Citcom's output data, files ending in .number are moved into directories called /step{number}/, and all q.number.dat files are moved into the q_step_dat directory.



Name

test2_step0.pdf
test2_step10.pdf
test2_step20.pdf

Figure 1: Generated temperature plots using command `TiCOM_bash last=20 postproc=postproc`.

1.3 Generated Plots

Although the generation of temperature plots using TiCOM output data & GMT (Generic Mapping Tools) is not a core functionality of TiCOM, there exists a means to easily generate these plots given the user has the necessary postprocessing scripts & the tools.

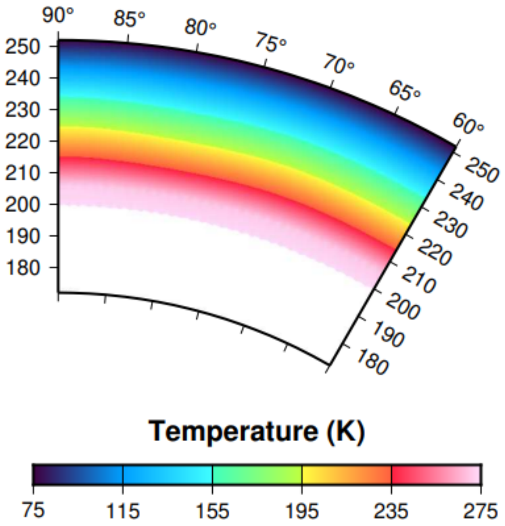


Figure 2: One of the temperature plots (not very much going on in this one).

Plots like the one shown above are not generated by default. If the user wants these plots to be generated, they should use the `postproc=dir` flag as specified in section 2. '`dir`' should refer to the name of the directory with the postprocessing scripts. In the current version of TiCOM, the supplied `dir` is called `/postproc/`. So, in that case, the user would run something like

```
user@host $ TiCOM_bash postproc=postproc
```

2 Usage at the Command-Line

TiCOM is a command-line tool, with the user script being written in Bash. Users can pass in flags at the command line when calling TiCOM to change variables that they would otherwise have to modify scripts for, saving time & effort. For a user script called `TiCOM_bash`, our usage is:

```
user@host $ TiCOM_bash                                       # Use defaults
user@host $ TiCOM_bash flag1 flag2 flag3...                  # Flags usage
user@host $ TiCOM_bash --m last=20 data="Desktop/output"  # Flags/options example
```

Note: If the shell cannot recognize your TiCOM script (eg. `TiCOM_bash`), add `./` to the beginning of the script name. This is to force the shell to look in your current directory for the script!

Category	Option/Flag	Description
Re-make	`--m`	Runs `make clean` + `make` on Citcom & TiRADE
	`--t`	Runs `make clean` + `make` on TiRADE
	`--c`	Runs `make clean` + `make` on Citcom

Help Menu	`--h`
	`--help`
	`help`
		Displays the help menu

Diagnostic Mode	`--d`	Provides an interface through which the user can configure certain TiCOM settings
Dimensionalize	`dimensionalize`	Dimensionalize Citcom data output to screen
Pathname Overload	`settings=file`	Provide alternate default settings file, set to ‘file’
	`data=dir`	Overload data directory default, set to ‘dir’
	`citcom=dir`	Overload Citcom source code directory default, set to ‘dir’
	`tirade=dir`	Overload TiRADE source code directory default, set to ‘dir’
	`citcom_in=file`	Overload Citcom input file default, load ‘file’
	`case=dir`	Overload CASE directory default, set to ‘dir’
	`postproc=dir`	Runs postprocessing tools after code TiCOM execution finished; looks for scripts under the directory ‘dir’
Step Overload	`first=x`	Set first step to integer x
	`last=x`	Set last step to integer x
	`inc=x`	Set step increment to integer x

Figure 3: Table of command-line arguments.

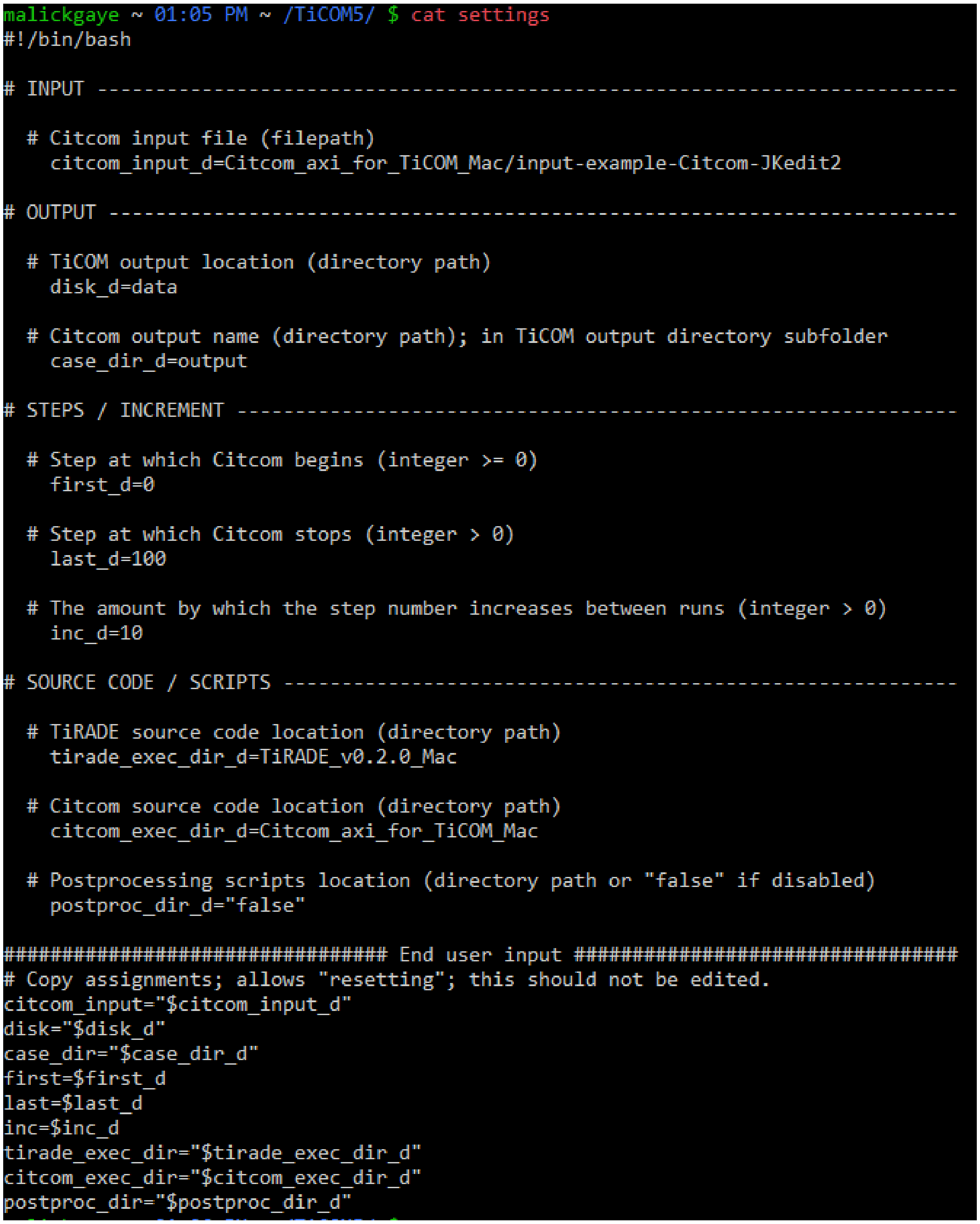
2.1 Default Settings/The ‘settings’ File

TiCOM uses default parameters every time that it is called. The utilization of some default parameters has been abstracted so that a user can easily provide or edit a file that TiCOM references to define these parameters upon initialization.

Please refer to the below list of variables that must be correctly assigned in order to run TiCOM:

- `disk_d` - Default data output directory
- `citcom_exec_dir_d` - Default Citcom source code directory
- `tirade_exec_dir_d` - Default TiRADE source code directory
- `case_dir_d` - Default Citcom CASE output directory
- `citcom_input_d` - Default Citcom input file path
- `postproc_directory_d` - Default postprocessing tools directory; “false” if disabled by default
- `first_d` - Default Citcom first step
- `last_d` - Default Citcom last step
- `inc_d` - Default Citcom step increment

Below those assignments, assignments from the defaults to the corresponding regular variables (without the ‘_d’ suffix) must be made. These should not change over time. So, a minimal default settings file would be appear as such:



```
malickgaye ~ 01:05 PM ~ /TiCOM5/ $ cat settings
#!/bin/bash

# INPUT ------------------------------------------------------------------------------

  # Citcom input file (filepath)
    citcom_input_d=Citcom_axi_for_TiCOM_Mac/input-example-Citcom-JKedit2

# OUTPUT -----------------------------------------------------------------------------

  # TiCOM output location (directory path)
    disk_d=data

  # Citcom output name (directory path); in TiCOM output directory subfolder
    case_dir_d=output

# STEPS / INCREMENT ------------------------------------------------------------------

  # Step at which Citcom begins (integer >= 0)
    first_d=0

  # Step at which Citcom stops (integer > 0)
    last_d=100

  # The amount by which the step number increases between runs (integer > 0)
    inc_d=10

# SOURCE CODE / SCRIPTS --------------------------------------------------------------

  # TiRADE source code location (directory path)
    tirade_exec_dir_d=TiRADE_v0.2.0_Mac

  # Citcom source code location (directory path)
    citcom_exec_dir_d=Citcom_axi_for_TiCOM_Mac

  # Postprocessing scripts location (directory path or "false" if disabled)
    postproc_dir_d="false"

################################# End user input #################################
# Copy assignments; allows "resetting"; this should not be edited.
citcom_input="$citcom_input_d"
disk="$disk_d"
case_dir="$case_dir_d"
first=$first_d
last=$last_d
inc=$inc_d
tirade_exec_dir="$tirade_exec_dir_d"
citcom_exec_dir="$citcom_exec_dir_d"
postproc_dir="$postproc_dir_d"
```

TiCOM will only correctly source the default settings if:

- The relevant file is titled `settings`, and is in the working directory of the user. For the sake of simplicity, if the user is running TiCOM from the TiCOM root folder in the terminal, then `settings` should be in that directory as well. **When in doubt, place the `settings` file in the same directory that the TiRADE/Citcom source code, etc. directories live in**

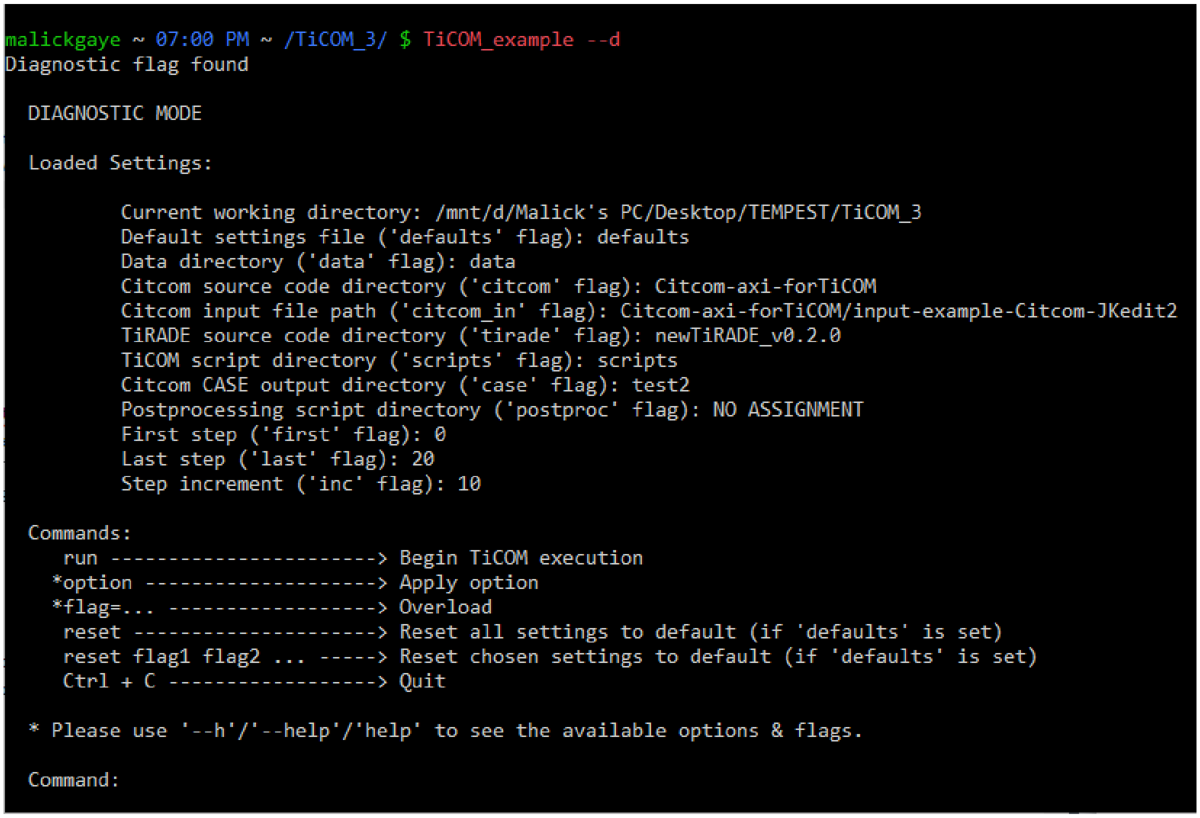
- Each of the listed variables are correctly assigned (`tirade_exec_dir_d` gives the location of

TiRADE source code, `citcom_input_d` is a Citcom input file, etc.)

The user can overload the settings that TiCOM (temporarily) uses by overloading using the `settings` flag name. At the command-line, overloading the default settings will reset settings with those given by the newly supplied file. This is why ensuring that this file is valid is important. Note that overloading TiCOM's settings with an alternate settings file *will not* change its default settings or default settings file permanently.

2.2 Diagnostic Mode

By using the `--d` option at the command-line when starting TiCOM, the user is presented an interface that will allow them configure some of TiCOM's settings in-terminal rather than editing the script itself. This allows temporary changes to TiCOM's configuration be made to quickly without having to edit any files.



```
malickgaye ~ 07:00 PM ~ /TiCOM_3/ $ TiCOM_example --d
Diagnostic flag found

  DIAGNOSTIC MODE

  Loaded Settings:

          Current working directory: /mnt/d/Malick's PC/Desktop/TEMPEST/TiCOM_3
          Default settings file ('defaults' flag): defaults
          Data directory ('data' flag): data
          Citcom source code directory ('citcom' flag): Citcom-axi-forTiCOM
          Citcom input file path ('citcom_in' flag): Citcom-axi-forTiCOM/input-example-Citcom-JKedit2
          TiRADE source code directory ('tirade' flag): newTiRADE_v0.2.0
          TiCOM script directory ('scripts' flag): scripts
          Citcom CASE output directory ('case' flag): test2
          Postprocessing script directory ('postproc' flag): NO ASSIGNMENT
          First step ('first' flag): 0
          Last step ('last' flag): 20
          Step increment ('inc' flag): 10

  Commands:
     run -----------------------> Begin TiCOM execution
    *option --------------------> Apply option
    *flag=... ------------------> Overload
     reset ---------------------> Reset all settings to default (if 'defaults' is set)
     reset flag1 flag2 ... -----> Reset chosen settings to default (if 'defaults' is set)
     Ctrl + C ------------------> Quit

  * Please use '--h'/'--help'/'help' to see the available options & flags.

  Command:
```

Figure 4: The interactive mode that appears when the user invokes --d; notice that the loaded settings are the default settings described in the previous section.

Entering nothing, `r`, or `run` starts TiCOM using the displayed settings. To exit without running TiCOM, use `Ctrl + C`; no permanent changes will be made. Options (usually denoted by the double hyphen `--`) can be entered and used normally here. The help menu can also be displayed from here if the user would like to see the available options and flags from the terminal rather than having to consult the documentation.

As well as the working directory that the user is in (which should be TiCOM root, as is displayed in the image above), the following [overload-able] settings (described by their flag names) are displayed to the screen:

- `settings` - Default settings file
- `data` - Data output directory
- `citcom` - Citcom source code directory

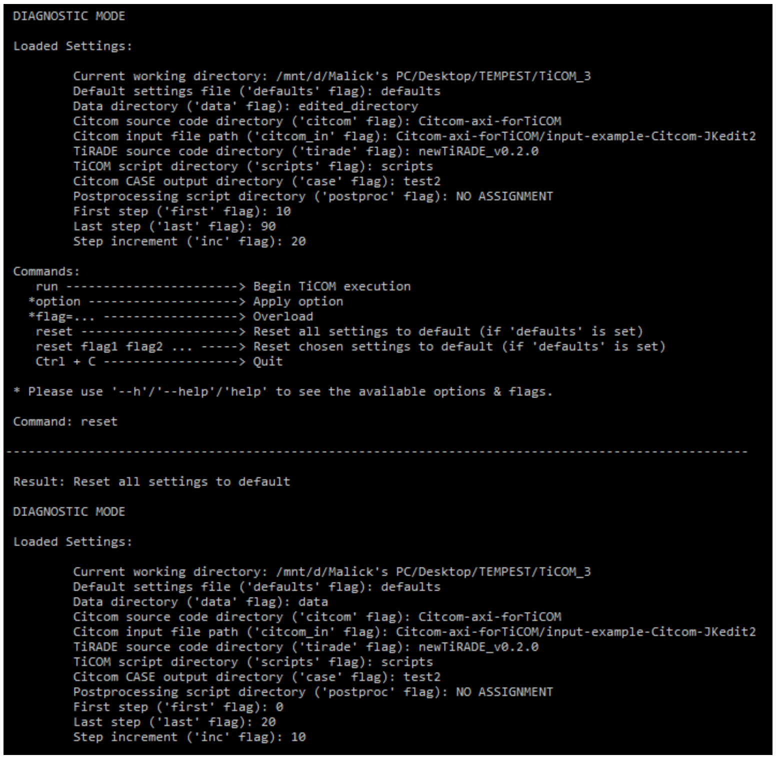
- `citcom_in` - Citcom input file path
- `tirade` - TiRADE source code directory
- `case` - Citcom CASE output directory
- `postproc` - TiCOM postprocessing script directory
- `first` - Step at which Citcom starts
- `last` - Start at which Citcom stops
- `inc` - The amount by which the step number increments

These can be overloaded at the interactive menu just as they can in the command line. The only *functional* difference in overloading here is the fact that in this mode, loading an alternate default settings file does not automatically reset all the parameters (whereas a reset is forced if a default settings file is loaded at the command-line).

Continuing on the topic of 'resetting' parameters, we have introduced a reset functionality that is specific to this diagnostic mode. While there is a default settings file loaded to TiCOM (in other words, when we do not see `NO ASSIGNMENT` printed as the loaded setting for it), the user can reset either all of the settings altogether, or chosen settings given by what is specified in the default settings file. Note that alternate or modified default settings files are not validated by TiCOM (only their existence is checked), so **it is important to ensure that variables are correctly assigned as described in the previous section**.

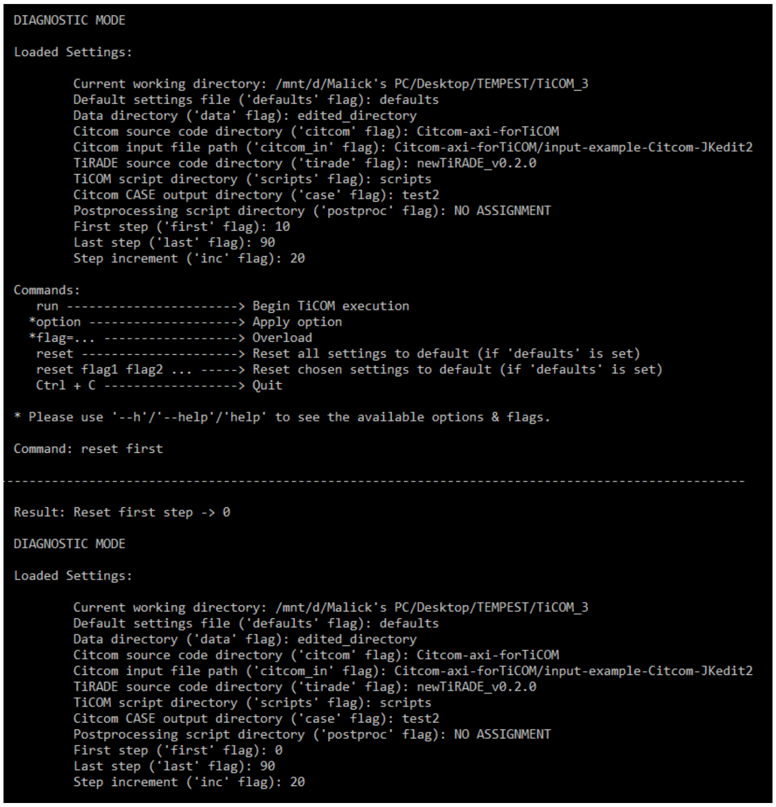
The user can either reset all settings by calling the `reset` command with no parameters, or reset settings attached to specific flags by calling the `reset` command with flag names following it, as is shown below.

```
Command: reset
Command: reset flag1 flag2 ...
```



```
DIAGNOSTIC MODE

Loaded Settings:

        Current working directory: /mnt/d/Malick's PC/Desktop/TEMPEST/TiCOM_3
        Default settings file ('defaults' flag): defaults
        Data directory ('data' flag): edited_directory
        Citcom source code directory ('citcom' flag): Citcom-axi-forTiCOM
        Citcom input file path ('citcom_in' flag): Citcom-axi-forTiCOM/input-example-Citcom-JKedit2
        TiRADE source code directory ('tirade' flag): newTiRADE_v0.2.0
        TiCOM script directory ('scripts' flag): scripts
        Citcom CASE output directory ('case' flag): test2
        Postprocessing script directory ('postproc' flag): NO ASSIGNMENT
        First step ('first' flag): 10
        Last step ('last' flag): 90
        Step increment ('inc' flag): 20

Commands:
   run ----------------------> Begin TiCOM execution
  *option -------------------> Apply option
  *flag=... -----------------> Overload
   reset --------------------> Reset all settings to default (if 'defaults' is set)
   reset flag1 flag2 ... -----> Reset chosen settings to default (if 'defaults' is set)
   Ctrl + C -----------------> Quit

* Please use '--h'/'--help'/'help' to see the available options & flags.

Command: reset

----------------------------------------------------------------------------------------------------

Result: Reset all settings to default

DIAGNOSTIC MODE

Loaded Settings:

        Current working directory: /mnt/d/Malick's PC/Desktop/TEMPEST/TiCOM_3
        Default settings file ('defaults' flag): defaults
        Data directory ('data' flag): data
        Citcom source code directory ('citcom' flag): Citcom-axi-forTiCOM
        Citcom input file path ('citcom_in' flag): Citcom-axi-forTiCOM/input-example-Citcom-JKedit2
        TiRADE source code directory ('tirade' flag): newTiRADE_v0.2.0
        TiCOM script directory ('scripts' flag): scripts
        Citcom CASE output directory ('case' flag): test2
        Postprocessing script directory ('postproc' flag): NO ASSIGNMENT
        First step ('first' flag): 0
        Last step ('last' flag): 20
        Step increment ('inc' flag): 10
```

Figure 5: Resetting without parameters



```
DIAGNOSTIC MODE

Loaded Settings:

        Current working directory: /mnt/d/Malick's PC/Desktop/TEMPEST/TiCOM_3
        Default settings file ('defaults' flag): defaults
        Data directory ('data' flag): edited_directory
        Citcom source code directory ('citcom' flag): Citcom-axi-forTiCOM
        Citcom input file path ('citcom_in' flag): Citcom-axi-forTiCOM/input-example-Citcom-JKedit2
        TiRADE source code directory ('tirade' flag): newTiRADE_v0.2.0
        TiCOM script directory ('scripts' flag): scripts
        Citcom CASE output directory ('case' flag): test2
        Postprocessing script directory ('postproc' flag): NO ASSIGNMENT
        First step ('first' flag): 10
        Last step ('last' flag): 90
        Step increment ('inc' flag): 20

Commands:
   run ----------------------> Begin TiCOM execution
  *option -------------------> Apply option
  *flag=... -----------------> Overload
   reset --------------------> Reset all settings to default (if 'defaults' is set)
   reset flag1 flag2 ... -----> Reset chosen settings to default (if 'defaults' is set)
   Ctrl + C -----------------> Quit

* Please use '--h'/'--help'/'help' to see the available options & flags.

Command: reset first

----------------------------------------------------------------------------------------------------

Result: Reset first step -> 0

DIAGNOSTIC MODE

Loaded Settings:

        Current working directory: /mnt/d/Malick's PC/Desktop/TEMPEST/TiCOM_3
        Default settings file ('defaults' flag): defaults
        Data directory ('data' flag): edited_directory
        Citcom source code directory ('citcom' flag): Citcom-axi-forTiCOM
        Citcom input file path ('citcom_in' flag): Citcom-axi-forTiCOM/input-example-Citcom-JKedit2
        TiRADE source code directory ('tirade' flag): newTiRADE_v0.2.0
        TiCOM script directory ('scripts' flag): scripts
        Citcom CASE output directory ('case' flag): test2
        Postprocessing script directory ('postproc' flag): NO ASSIGNMENT
        First step ('first' flag): 0
        Last step ('last' flag): 90
        Step increment ('inc' flag): 20
```

Figure 6: Resetting with parameters

2.3 Re-making TiRADE/Citcom Source Code

Both Citcom & TiRADE have their own individual `Makefile`s that must be used at some point for the Citcom/TiRADE binaries to exist. Both of these programs must be compiled before we run TiCOM; the `--m` flag ensures that the user has recently-compiled versions of Citcom/TiRADE before running the main TiCOM loop, and is recommended for this reason.

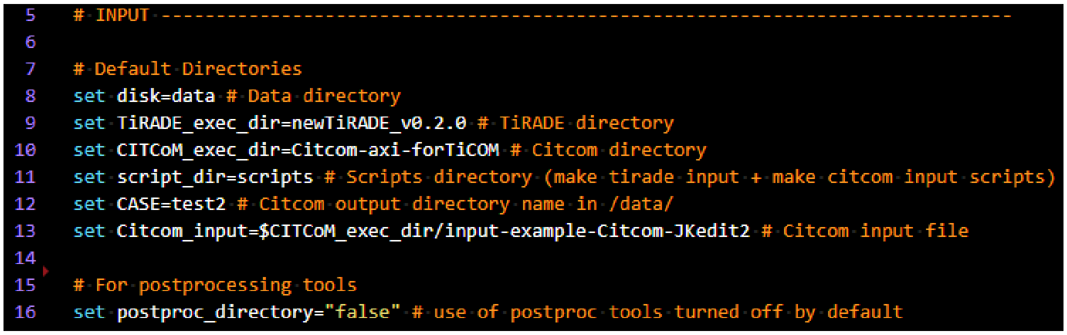
TiCOM assumes that the user has a `Makefile` in the Citcom & TiRADE executable directories, in case the user wants to use any of the re-`make` options. If no `make clean` or `make tirade.x/citcom.x` rule is found, or if the `Makefile`(s) do not exist, TiCOM will have an error. There should be a working `Makefile` for Citcom & TiRADE pre-supplied, and the following flags can be used to trigger the `make` rules:

- `--m` (`make clean` + `make` Citcom & TiRADE)
- `--c` (`make clean` + `make` Citcom)
- `--t` (`make clean` + `make` TiRADE)

2.4 Directories & Pathname Overloading

TiCOM is intended to be modular. Due to this there are capabilities to overload pathname defaults specified in the TiCOM script. Below is a table of the relevant variable names in the TiCOM script, how these variables are overloaded at the command line, their default settings and their description.

Variable Name	**Overload Format**	**Default**	**Description**
`settings_file`	`settings=file`	`./settings`	File that makes variable assignments that TiCOM loads upon initialization.
`disk`	`data=dir`	`./data/`	The directory to which all generated TiCOM files go. Contains the subdirectories `/Citcom_data/`, `/Tirade_data/`, and `./temp_plots/` (if the use of postprocessing tools was requested using the '`postproc=dir`' flag).
`tirade_exec_dir`	`tirade=dir`	`./TiRADE_v0.2.0/`	The directory where the TiRADE source code lives.
`citcom_exec_dir`	`citcom=dir`	`./Citcom-axi-forTiCOM/`	The directory where the Citcom source code lives.
`case`	`case=dir`	`./test2/`	The subdirectory in the `./data/Citcom_data/` directory where generated Citcom data is output.
`citcom_input`	`citcom_in=file`	`./Citcom-axi-forTiCOM/input-example-Citcom-JKedit2`	The directory where the initial Citcom input file lives.
`postproc_directory`	`postproc=dir`	`"false"` (Default off or `--np`) `postproc` or other (`--p`)	The directory where the postprocessing scripts live.



```
# INPUT ---------------------------------------------------------------------------------

# Default Directories
set disk=data # Data directory
set TiRADE_exec_dir=newTiRADE_v0.2.0 # TiRADE directory
set CITCoM_exec_dir=Citcom-axi-forTiCOM # Citcom directory
set script_dir=scripts # Scripts directory (make tirade input + make citcom input scripts)
set CASE=test2 # Citcom output directory name in /data/
set Citcom_input=$CITCoM_exec_dir/input-example-Citcom-JKedit2 # Citcom input file

# For postprocessing tools
set postproc_directory="false" # use of postproc tools turned off by default
```

Figure 7: Default directories/pathnames in the TiCOM user script.

The default settings are defined in the `settings` file. If one wishes to change their default parameters permanently, it should be done there.

2.5 Step Overloading

There are three step-related variables in TiCOM:

- `first`, the step at which TiCOM starts running; default 0.
- `last`, the step at which TiCOM stops running; default 100.
- `inc`, the amount by which the current step increases between iterations; default 10.

To overload the relevant step variable directory the user can use the following at the command line or in the diagnostic:

- `first=X` $(X \geq 0)$
- `last=X` $(X > 0)$
- `inc=X` $(X > 0)$

2.6 Additional Notes

- Default settings should be changed in the `settings` file.
- Some of the scripts are written in csh, if the syntax looks unfamiliar.
  - csh is very fidgety - unless you are attempting to change TiCOM's functionality then it is recommended to change only the defaults (first ≈ 25 lines of the TiCOM user script).
  - Generally, we should avoid overwriting files that the user gives to us - when in doubt, add a command to the script to copy/paste the file in question to a copy that we can then copy. For example, if the user specifies an alternate output data directory with '`data=dir`', TiCOM must update the listed output directory in the initial Citcom input file in order to output the data in the correct spot. Instead of using `sed` to change the

pathname in the user-supplied initial Citcom input file, we copy the initial Citcom input file and change *that* instead. This way the user can continue to keep using the file they gave to us.

- TiCOM's functions can be used independently of TiCOM, by sourcing the `functions` file.

3 Miscellaneous

3.1 Bugs

See the next section for TODOs on the clearly identified causes of various bugs. Bugs that have an unclear cause are listed below, along with the general time period of discovery.

- December 2021: Compilation warnings upon running `make citcom.x` & `make tirade.x` (non-critical).

- January 2022: Known bugs when the user uses pathnames with spaces in them. For PC & Linux, the use of spaces in pathnames is currently not advised. On Mac, spaces can be used in pathnames without issue.

3.2 Future Work

- If the postprocessing scripts don't refer to the same data location as the `disk` variable then TiCOM should copy these scripts and use `sed` to change the location of the output data.

- Implementation of a 'true' fluid layer (i.e. alternate propagators through liquid layers).

4 Contact

Name	**Email Address**
James Roberts	james.roberts@jhuapl.edu
Person 2	email 2
Person 3	email 3
Person 4	email 4
Malick Gaye	mgaye3@jhu.edu

TODO: Additional emails / Other additional contact info