



# KTH Framework; multiple tools for CFD modelling



ROYAL INSTITUTE  
OF TECHNOLOGY

**Adam Peplinski**  
KTH Engineering Mechanics



# Outlook

---



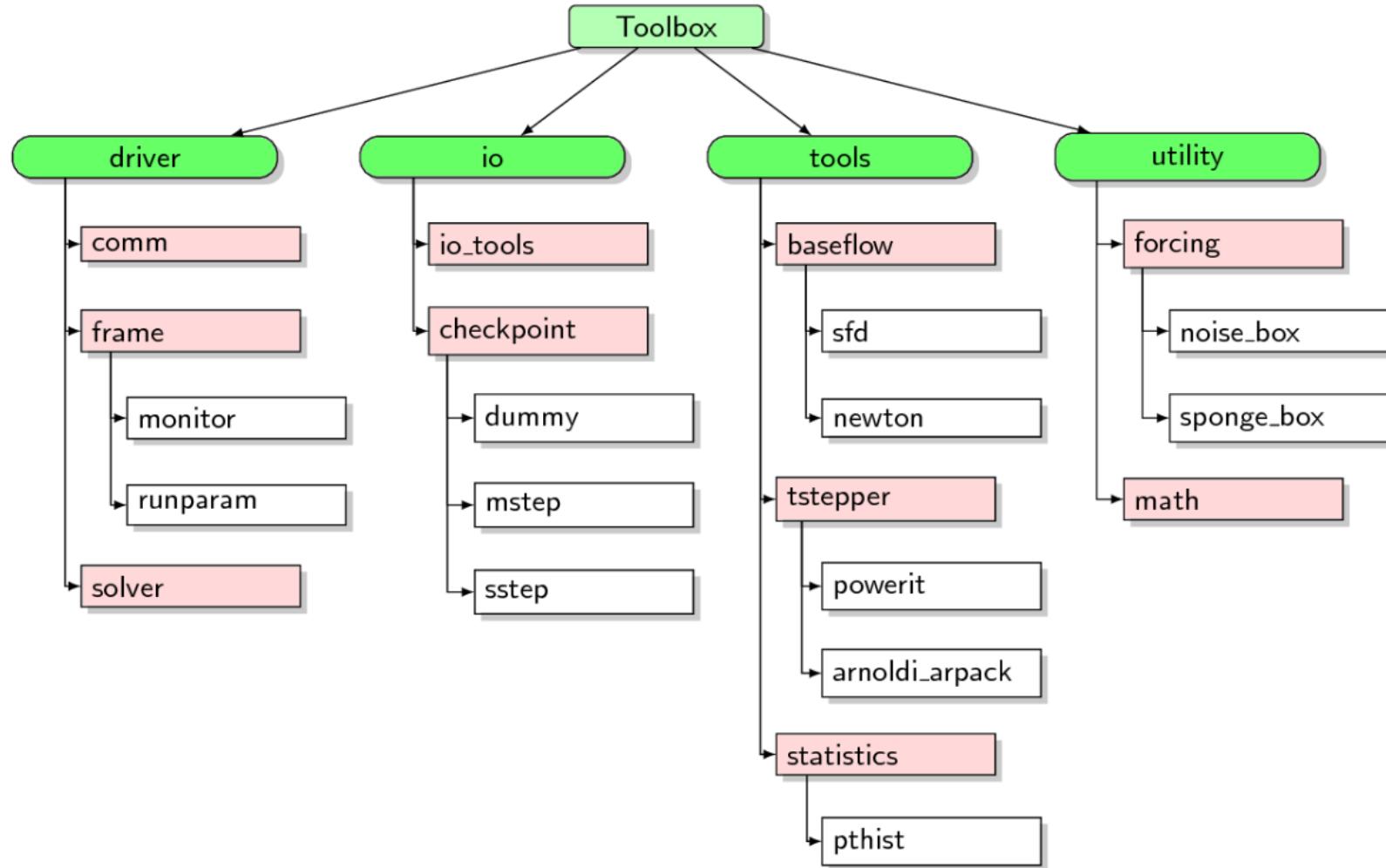
- Framework idea and structure
- Doxygen documentation
- Setup a case

# Framework



- Set of routines that would allow to build a set of non-conflicting tools
- Nonintrusive; operates through existing user interface (no core code changes)
- Follow Nek5000 philosophy
- Flexible; easy to expand and modify
- Well documented with the code available online (doxygen)

# Framework structure



<https://github.com/KTH-Nek5000>



KTH Mechanics

https://www.kth.se/s...

Overview    Repositories 22    Packages    People 16    Teams 4    Projects    Settings

Pinned

KTH\_Framework    Framework for toolboxes developed for Nek5000    7 ⭐ 5 ⚡

KTH\_Examples    Python 5 ⭐ 6 ⚡

KTH\_Toolbox    Fortran 7 ⭐ 10 ⚡

Customize your pins

People

Invite someone

Repositories

Find a repository...    Type    Language    Sort    New

KTH\_Examples

Python 5 ⭐ 6 ⚡ 0 ⚡ 0    Updated 5 hours ago

KTH\_Framework

ENCCS Nek5000 training 2021

# <https://github.com/KTH-Nek5000>



master	2 branches	0 tags	Go to file	Add file	Code
adampep Updated PSTAT2D	f4c8069 6 hours ago	87 commits			
Examples @ 89747bb Updated PSTAT2D	6 hours ago				
Nek5000 @ 9b3d922 Number of updates	12 months ago				
Toolbox @ f1bee5d Added Readme and small documentation modi...	21 hours ago				
docs Framework update	3 years ago				
.gitignore Added doxygen files	5 years ago				
.gitmodules Adding examples submodule	5 years ago				
FRAMEWORK.txt Added doxygen files	5 years ago				
README.md Minimalistic README file	23 hours ago				

## README.md

# KTH Framework

## About

Framework for toolboxes developed for Nek5000

[kth-nek5000.github.io/k...](https://kth-nek5000.github.io/k...)

Readme

## Releases

No releases published  
[Create a new release](#)

## Packages

No packages published  
[Publish your first package](#)

## Contributors 2

adampep Adam Poplin

# KTH framework for Nek5000 toolboxes; testing version 0.0.1

Main Page	Related Pages	Modules	Data Types List ▾	Files ▾	 
-----------	---------------	---------	-------------------	---------	---

✓ KTH framework for Nek5000 toolboxes; test

- ▶ Framework for toolboxes developed for Nek5000.
- ▶ Runtime parameters for KTH framework.

Setting up the case.

Tool development workflow.

List of interfaces provided by modules.

List of interfaces required by modules.

Todo List

Bibliography

- ▶ Modules
- ▶ Data Types List
- ▶ Files

• Nek5000,  
• toolbox,  
• tool examples.

Each of these repositories can be downloaded separately by executing

```
git clone https://github.com/Nek5000/Nek5000.git -b master
git clone https://github.com/KTH-Nek5000/KTH_Toolbox -b master
git clone https://github.com/KTH-Nek5000/KTH_Examples -b master
```

or as a single framework by

```
git clone --recursive https://github.com/KTH-Nek5000/KTH_Framework
```

The second method is preferred as Nek5000 repository has neither development nor stable branch

# KTH framework for Nek5000 toolboxes; testing version 0.0.1

Main Page	Related Pages	Modules	Data Types List ▾	Files ▾	 
-----------	---------------	---------	-------------------	---------	---

✓ KTH framework for Nek5000 toolboxes; test

- ▶ Framework for toolboxes developed for Nek5000.
- ▶ Runtime parameters for KTH framework.

Setting up the case.

Tool development workflow.

List of interfaces provided by modules.

List of interfaces required by modules.

Todo List

Bibliography

- ▶ Modules
- ▶ Data Types List
- ▶ Files

• Nek5000,  
• toolbox,  
• tool examples.

Each of these repositories can be downloaded separately by executing

```
git clone https://github.com/Nek5000/Nek5000.git -b master
git clone https://github.com/KTH-Nek5000/KTH_Toolbox -b master
git clone https://github.com/KTH-Nek5000/KTH_Examples -b master
```

or as a single framework by

```
git clone --recursive https://github.com/KTH-Nek5000/KTH_Framework
```

The second method is preferred as Nek5000 repository has neither development nor stable branch



```
!=====
subroutine userchk
implicit none
include 'SIZE'
include 'TSTEP'          ! ISTEP, lastep, time
include 'INPUT'           ! IF3D, PARAM
!
if (istep.eq.0) then
start framework
  call frame_start
!
set volume flow parameters
  param(54) = uparam(1)
  param(55) = uparam(2)
endif
!
monitor simulation
call frame_monitor
!
for statistics
call stat_avg
!
finalise framework
if (istep.eq.nsteps.or.lastep.eq.1) then
  call frame_end
endif
!
return
end
=====
```

```
!=====
subroutine frame_usr_register
implicit none
!
include 'SIZE'
include 'FRAMELP'
!
call stat_register
return
end subroutine
=====
subroutine frame_usr_init
implicit none
!
include 'SIZE'
include 'FRAMELP'
!
call stat_init
return
end subroutine
=====
subroutine frame_usr_end
implicit none
!
include 'SIZE'
include 'FRAMELP'
!
call stat_end
return
end subroutine
=====
=====
```

# Runtime parameters



KTH framework for Nek5000 toolboxes; test

▶ Framework for toolboxes developed for N

▶ Runtime parameters for KTH framework.

Setting up the case.

Tool development workflow.

List of interfaces provided by modules.

List of interfaces required by modules.

Todo List

Bibliography

▶ Modules

▶ Data Types List

▶ Files

```
-----  
residualProj = yes  
  
[VELOCITY]  
residualTol = 1e-8  
residualProj = yes  
density = 1.0  
viscosity = -2.5E+03  
advection = yes  
  
#  
[_RUNPAR]           # Runtime parameter section  
PARFWRITE          = no                      # Do w  
PARFNAME            = outparfile             # Runt  
                  .par)  
#  
[_MONITOR]          # Runtime parameter section  
LOGLEVEL            = 4                       # Logg  
WALLTIME            = 23:45                  # Simu  
#  
[_STAT]              # Runtime paramere section for s  
AVSTEP               = 10  
IOSTEP                = 50
```

# Runtime parameters



Main Page	Related Pages	Modules	Data Types List ▾	Files ▾																					
KTH framework for Nek5000 toolboxes; test ▶ Framework for toolboxes developed for Nek5000 ▼ Runtime parameters for KTH framework. Runtime parameter file. Module's runtime parameters. List of runtime parameters provided by modules. List of runtime parameters required by modules. Setting up the case. Tool development workflow. List of interfaces provided by modules. List of interfaces required by modules. Todo List Bibliography ▶ Modules ▶ Data Types List ▶ Files																									
<h3>Module chkpoint</h3> <p>Parameters provided by checkpoint module (include file - <i>CHKPOINT.D</i>):</p> <table border="1"><thead><tr><th>Variable</th><th>Type</th><th>Runtime parameter</th><th>Description</th></tr></thead><tbody><tr><td>chpt_ifrst</td><td>logical</td><td>_chkpoint:readchkpt</td><td>start simulation from checkpoint</td></tr><tr><td>chpt_fnum</td><td>integer</td><td>_chkpoint:chkpFNumber</td><td>restart file number</td></tr><tr><td>chpt_step</td><td>integer</td><td>_chkpoint:chkpInterval</td><td>checkpoint dump frequency (number of time steps)</td></tr></tbody></table>						Variable	Type	Runtime parameter	Description	chpt_ifrst	logical	_chkpoint:readchkpt	start simulation from checkpoint	chpt_fnum	integer	_chkpoint:chkpFNumber	restart file number	chpt_step	integer	_chkpoint:chkpInterval	checkpoint dump frequency (number of time steps)				
Variable	Type	Runtime parameter	Description																						
chpt_ifrst	logical	_chkpoint:readchkpt	start simulation from checkpoint																						
chpt_fnum	integer	_chkpoint:chkpFNumber	restart file number																						
chpt_step	integer	_chkpoint:chkpInterval	checkpoint dump frequency (number of time steps)																						
<h3>Module monitor</h3> <p>Parameters provided by monitor module (include file - <i>MNTRLOG.D</i>):</p> <table border="1"><thead><tr><th>Variable</th><th>Type</th><th>Runtime parameter</th><th>Default value</th><th>Description</th></tr></thead><tbody><tr><td>mntr_lp_def</td><td>integer</td><td>_monitor:loglevel</td><td>lp_inf</td><td>Logging threshold for toolboxes</td></tr><tr><td>mntr_iftdsc</td><td>logical</td><td>_monitor:iftimdscr</td><td>.false.</td><td>Write timer description in the summary</td></tr><tr><td>mntr_wtimes</td><td>string</td><td>_monitor:WallTime</td><td>00:00</td><td>simulation wall time</td></tr></tbody></table>						Variable	Type	Runtime parameter	Default value	Description	mntr_lp_def	integer	_monitor:loglevel	lp_inf	Logging threshold for toolboxes	mntr_iftdsc	logical	_monitor:iftimdscr	.false.	Write timer description in the summary	mntr_wtimes	string	_monitor:WallTime	00:00	simulation wall time
Variable	Type	Runtime parameter	Default value	Description																					
mntr_lp_def	integer	_monitor:loglevel	lp_inf	Logging threshold for toolboxes																					
mntr_iftdsc	logical	_monitor:iftimdscr	.false.	Write timer description in the summary																					
mntr_wtimes	string	_monitor:WallTime	00:00	simulation wall time																					
<h3>Module noise_box</h3> <p>Parameters provided by sponge_box module (include file - <i>SPONGRBD.D</i>):</p>																									

# Interface list



Main Page	Related Pages	Modules	Data Types List ▾	Files ▾
▼ KTH framework for Nek5000 toolboxes; test <ul style="list-style-type: none"><li>▶ Framework for toolboxes developed for Nek5000.</li><li>▶ Runtime parameters for KTH framework.</li></ul>			<ul style="list-style-type: none"><li>o. <a href="#">io_mfios</a></li><li>9. <a href="#">io_mfiv</a></li><li>10. <a href="#">io_mfis</a></li></ul>	
<h3>Module math</h3> <p>Interface provided:</p> <ul style="list-style-type: none"><li>1. <a href="#">mth_stepf</a></li><li>2. <a href="#">mth_rand</a></li></ul>				
<h3>Module monitor</h3> <p>Interface provided:</p> <ul style="list-style-type: none"><li>1. <a href="#">mntr_set_step_delay</a></li><li>2. <a href="#">mntr_get_step_delay</a></li><li>3. <a href="#">mntr_set_conv</a></li><li>4. <a href="#">mntr_is_initialised</a></li><li>5. <a href="#">mntr_lp_def_get</a></li><li>6. <a href="#">mntr_mod_reg</a></li><li>7. <a href="#">mntr_set_conv</a></li></ul>				

# List of tools



Main Page | Related Pages | Modules

Data Types List | Files

- ▼ KTH framework for Nek5000 toolboxes; testing
  - Framework for toolboxes developed for Nek5000
  - Runtime parameters for KTH framework.
    - Setting up the case.
    - Tool development workflow.
  - List of interfaces provided by modules.
  - List of interfaces required by modules.
  - Todo List
  - Bibliography
- Modules
- Data Types List
- Files

## Modules

Here is a list of all modules:

▼ Framework for Nek5000 toolboxes	Framework for KHT toolboxes
▼ Toolbox	Toolbox for Nek5000
▼ Driver module	Driver routines for toolboxes
Communication routines.	MPI communication wrappers
▼ Frame module	Backbone for toolboxes
Monitoring module	Monitoring routines for toolboxes
Runtime parameters	Routines related to module's runtime parameters
▼ Input/output module	I/O routines for toolboxes
▼ Checkpointing routines	Checkpointing routines for toolbox
Dummy checkpointing	Dummy routines for checkpointing
Multistep checkpointing	Multistep checkpointing for DNS, MHD and linear simulations
I/O tools	Input/Output simple tools
▼ Post processing tools	Set of tools for post-processing
Post-processing of 2D statistics	Set of post-processing routines of 2D statistics
Post-processing of 3D statistics	Set of post-processing routines for 3D statistics
▼ Tools module	Set of tools for Nek5000 framework
▼ Base flow calculation	Set of tools for base flow calculation
SFD module	Set of routine for Selective Frequency Damping

# Checkpointing tool



ing  
ek5

## Checkpointing routines

Framework for Nek5000 toolboxes » Toolbox » Input/output module

Checkpointing routines for toolbox. [More...](#)

Collaboration diagram for Checkpointing routines:

```
graph LR; IO[Input/output module] <--> CR[Checkpointing routines]; CR <--> DC[Dummy checkpointing]; CR <--> MC[Multistep checkpointing]
```

**Modules**

- Dummy checkpointing**  
Dummy routines for checkpointing.
- Multistep checkpointing**  
Multistep checkpointing for DNS, MHD and linear simulations.

**Files**

- file `chkpoint.f`**  
Set of checkpoint routines.

**Functions**

# Checkpointing tool



## Detailed Description

Checkpointing routines for toolbox.

This is the main interface to I/O checkpointing.

### Module interface:

#### Global interface list:

Interface provided:

1. `chkpt_register`
2. `chkpt_init`
3. `chkpt_main`
4. `chkpt_get_fset`

#### Global interface dependency:

Interface required:

1. `chkpts_register`
2. `chkpts_init`
3. `chkpts_read`
4. `chkpts_write`

### Module interface usage:

```
!=====
      subroutine userchk
      include 'TSTEP'

      if (istep.eq.0) then
      start framework
      call frame_start
      endif

      ! monitor simulation
      call frame_monitor

      ! save/load files for full-restart
      call chkpt_main

      ! finalise framework
      if (istep.eq.nsteps.or.lastep.eq.1) then
      call frame_end
      endif

      return
      end

!=====
      subroutine frame_usr_register
      implicit none

      include 'SIZE'
      include 'FRAMELP'

      ! register modules
      call io_register
      call chkpt_register
```

# Checkpointing tool



ng  
k5

```
end subroutine
!=====
Module parameters:
```

Global parameter list:

Parameters provided by checkpoint module (include file - *CHKPOINTD*):

Variable	Type	Runtime parameter	Description
chpt_ifrst	logical	_chkpoint:readchkpt	start simulation from checkpoint
chpt_fnum	integer	_chkpoint:chkpFnumber	restart file number
chpt_step	integer	_chkpoint:chkpInterval	checkpoint dump frequency (number of time steps)

Module parameter usage:

```
[_CHKPOINT]
chkpInterval = 10
readchkpt = no
chkpFnumber = 1
```

Function Documentation

## Function Documentation

◆ **chkpt\_get\_fset()**

```
subroutine chkpt_get_fset ( integer step_cnt,
                           integer set_out
                         )
```

Get step count to the checkpoint and a set number.

**Parameters**

- [out] **step\_cnt** decreasing step count in checkpoint writing phase (otherwise -1)
- [out] **set\_out** set number

Definition at line 256 of file **chkpoint.f**.

▼ Here is the caller graph for this function:

```
graph TD; sfd_main[sfd_main] --> sfd_RST_write[sfd_RST_write]; stepper_main[stepper_main] --> stepper_WRITE[stepper_WRITE]; stepper_VSOLVE[stepper_VSOLVE] --> stepper_WRITE; gsem_main[gsem_main] --> gsem_RST_write[gsem_RST_write]; stepper_WRITE --> chkpt_get_fset[chkpt_get_fset]; gsem_RST_write --> chkpt_get_fset;
```

# Checkpointing tool



- ▶ Framework for toolboxes developed for Nek
- ▶ Runtime parameters for KTH framework.
- Setting up the case.
- Tool development workflow.
- List of interfaces provided by modules.
- List of interfaces required by modules.
- Todo List
- Bibliography
- ▶ Modules
- ▶ Data Types List
- ▼ Files
  - ▼ File List
    - ▶ Nek5000
    - ▼ Toolbox
      - ▶ driver
      - ▼ io
        - ▼ checkpoint
          - ▶ dummy
          - ▶ mstep
          - ▶ chkpoint.f
    - ▶ io\_tools
    - ▶ postprocessing
    - ▶ tools
    - ▶ utility
  - ▶ Globals

```
194      end function
195 !=====
199 ! after frame monitor
200 subroutine chkpt_main
201 implicit none
202
203 include 'SIZE'
204 include 'TSTEP'
205 include 'CHKPOINTD'
206
207 ! local variables
208 integer itmp, lstdl
209 !-----
210 if(chpt_ifrst.and.istep.le.chpt_istep) then
211   call chkpts_read
212 elseif (istep.gt.chpt_istep) then
213
214   ! adjust max ISTEP for cyclic checkpoint writing
215   call mntr_get_step_delay(lstdl)
216   chpt_nstep = nsteps - lstdl - 1
217   ! check if chpt_nstep is in the middle of writing cycle
218   itmp = chpt_nstep + chpt_step - 1
219   if (mod(itmp,chpt_step).ge.(chpt_step-lstdl)) then
220     itmp = lstdl + mod(itmp,chpt_step) + 1 - chpt_step
221     chpt_nstep = chpt_nstep - itmp
222   endif
223
224   ! count steps to the end of writing stage
225   itmp = istep + chpt_step - 1
226   if (istep.gt.(nsteps-lstdl)) then
227     chpt_stepc = nsteps-istep+1
228   elseif (istep.lt.chpt_nstep.and.
229     mod(itmp,chpt_step).ge.(chpt_step-lstdl)) then
230     chpt_stepc = chpt_step - mod(itmp,chpt_step)
231   else
232     chpt_stepc = -1
233   endif
```

List of interfaces provided by modules. 

List of interfaces required by modules.

Todo List

Bibliography

► Modules

► Data Types List

▼ Files

▼ File List

▼ Nek5000

► 3rd\_party

▼ core

► 3rd\_party

► experimental

► bdry.f

► byte.c

► byte\_mpi.f

► calcz.f

► chelpers.c

► coef.f

► comm\_mpi.f

► conduct.f

► connect1.f

► connect2.f

► convect.f

► convect2.f

## bdry.f File Reference

Go to the source code of this file.

### Functions/Subroutines

subroutine **setlog** (ifecho)

subroutine **setrzer**

subroutine **chknord** (IFALGN, IFNORX, IFNORY, IFNORZ, IFC, IEL)

subroutine **chkaxcb**

subroutine **chkcbc** (CB, IEL, IFC, IFALGN, IERR)

subroutine **bcmask**

subroutine **bcdirvc** (V1, V2, V3, mask1, mask2, mask3)

subroutine **bcdirsc** (S)

subroutine **bcneusc** (S, ITYPE)

subroutine **faceis** (CB, S, IEL, IFACE, NX, NY, NZ)

subroutine **faceiv** (CB, V1, V2, V3, IEL, IFACE, NX, NY, NZ)

subroutine **nekasgn** (ix, iy, iz, e)

subroutine **bcneutr**

subroutine **trcon** (TRX, TRY, TRZ, TR1, TR2, TR3, IEL, IFC)

# Multistep checkpointing



## Detailed Description

Multistep checkpointing for DNS, MHD and linear simulations.

In the case of multistep restart one needs data from NBDINP timestep, where NBDINP is time integration order. I adapt **full\_restart\_save** subroutines for DNS, MHD and perturbation simulations. There are three possible sets of restart files:

1. 'rsX...' storing DNS data,
2. 'rbX...' storing MHD data,
3. 'rpX...' storing perturbation data,

where X is equal to 2\*NBDINP and gives a max number of restart files. Only NBDINP files is required for restart and the overwriting. Checkpoints are saved in double precision in binary ###.f format.

The possible combinations of files are:

1. 'rsX...' alone for DNS mode,
2. 'rsX...' and 'rbX...' for MHD mode,
3. 'rsX...' and 'rpX...' for perturbation mode.

In the perturbation mode only the first perturbation is written to the checkpoint file. To save disc space the mesh data is not stored. For not moving boundaries (IFMVBD=FALSE) only the first 'rsX...' file in the set contains mesh data. In the similar way for fixed base flow (IFBASE=FALSE) only single 'rsX...' file is generated.

# Toolbox



KTH-Nek5000/KTH\_Toolbox

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

f1bee5db40 1 branch 0 tags Go to file Code About

timofeymukha Small documentation upd... f1bee5d 23 hours ago 176 commits

driver	Fix some typos/missing articles in FRAMEWO...	6 months ago
io	Small correction of checkpointing code descript...	2 months ago
postprocessing	Small documentation updates (#8)	23 hours ago
tools	Small documentation updates (#8)	23 hours ago
utility	Updated compilation workflow	last month
.gitignore	Starting project and adding gitignore	5 years ago
FRAMEWORK.txt	Small documentation updates (#8)	23 hours ago
README.md	Create README.md	yesterday
toolbox_path.sh	Updated compilation workflow	last month

README.md

Readme

No description, web or topics provided.

Readme

Releases

No releases published Create a new release

Packages

No packages published Publish your first package

Contributors 3

adampep Adam P

timofeymukha Tim



# Examples



KTH-Nek5000/KTH\_Examples

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

89747bb931 1 branch 0 tags Go to file Code

salrm8	small corrections	89747bb 16 hours ago	101 commits
ext_cyl_ARN	Corrected typo in a compilation script	last month	
ext_cyl_DNS	Corrected typo in a compilation script	last month	
ext_cyl_LIN	Corrected typo in a compilation script	last month	
ext_cyl_PWI	Corrected typo in a compilation script	last month	
ext_cyl_SFD	Corrected typo in a compilation script	last month	
phill_PSTAT2D	small corrections	16 hours ago	
phill_PSTAT3D	Renamed files/routines to split pstat2d and p... 2 days ago		
phill_STAT	Corrected typo in a compilation script	last month	
phill_TRIP	Corrected typo in a compilation script	last month	
phill_TSRS	Corrected typo in a compilation script	last month	

About  
No desc or topics  
Readme

Release  
No release  
Create a new

Packages  
No packages  
Publish your

Contributors

# Examples



Each example consists of:

- required tools source and include files
- setup source file `###.usr`
- required `SIZE` file containing definitions of static arrays dimensions
- compilation script `compile_script`
- `makefile_usr.inc` describing compilation rules for tools files
- runtime parameters file `###.par`
- mesh description; it can be `###.box` or `###.re2`
- `README.md` listing all the tools

To compile the code:

- make sure `compile_script` has set execute bit (`chmod u+x compile_script`)
- edit `compile_script` to set proper compilers
- if you have not cloned this repo as a submodule of `KTH_Framework`, make sure that in the `compile_script` the `TOOLBOX_SRC` variable points to the root of KTH\_Toolbox, and `NEK_SOURCE_ROOT` points to the root of Nek5000.
- clean directory `./compile_script --clean` (with the first compilation on a given machine clean the libraries as well)
- build the code `./compile_script --all`

# Examples



## Example list

- `ext_cyl_DNS` - nonlinear simulation of the flow past circular cylinder
- `ext_cyl_LIN` - linear simulation of the flow past circular cylinder
- `ext_cyl_SFD` - nonlinear simulation of the flow past circular cylinder with Selective Frequency Damping to calculate base flow
- `ext_cyl_ARN` - linear stability tool for spectra calculation with Arnoldi algorithm
- `ext_cyl_PWI` - linear stability tool, power iteration
- `phill_STAT` - nonlinear simulation of flow over periodic hill to calculate turbulence statistics
- `phill_PSTAT` - postprocessing of files produced by `phill_STAT`
- `phill_TRIP` - nonlinear simulation of flow over periodic hill with tripping line