



BETTER SHIPS, BLUE OCEANS

# SEACAL

## Cross Platform Parallel Hybrid Code for Calculations of Ship Motions

PhD Sebastian Costamagna  
Scientific Software Specialist  
R&D Dept. - MARIN  
The Netherlands

## MARITIME RESEARCH INSTITUTE NETHERLANDS

BETTER SHIPS, BLUE OCEANS

MARIN is a globally recognised top institute for hydrodynamic and nautical research. Our mission is 'Better Ships, Blue Oceans': we stand for clean, smart and safe shipping and sustainable use of the sea. We do this as an independent knowledge partner for the maritime sector, government and society.

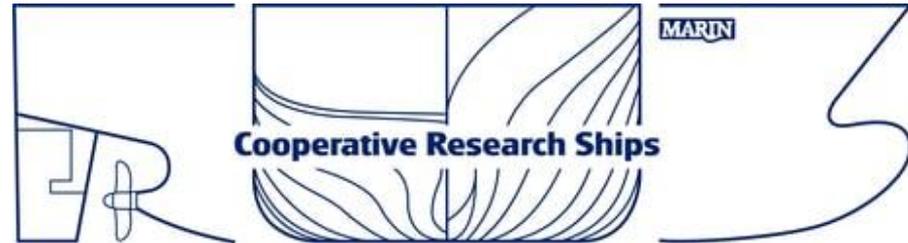


## SEAkeeping CALculations

### User group

>50 Users

>15 Institutions



### Dev. Team

-T. Bunnik

-E. van Dalen

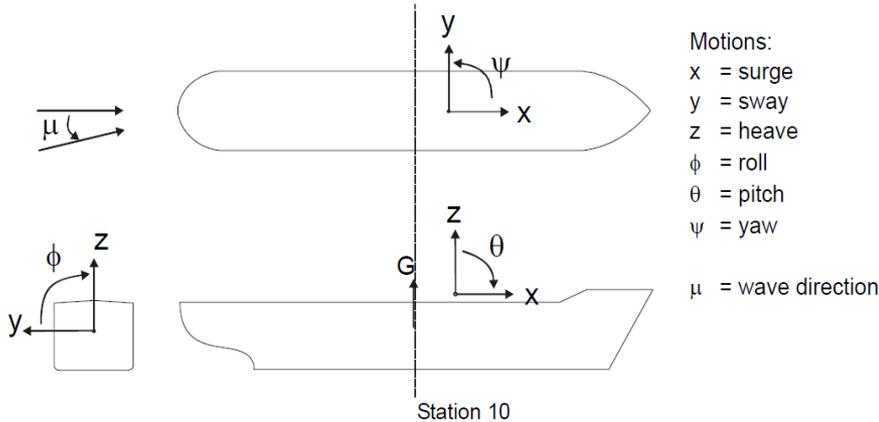
-R. van 't Veer

-S. Costamagna

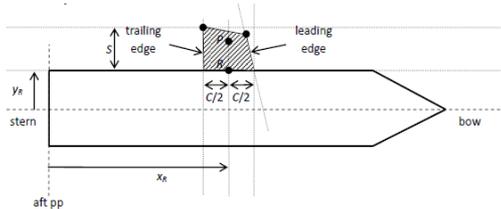


Frequency domain linear 3D diffraction theory. Advanced methods for calculations of roll damping, multi-hull/bodies motions, flexible ship responses, drift forces, effect of appendages, etc. on regular waves

## Wave direction and motions



## Appendages



## Zero-speed Green's function Method

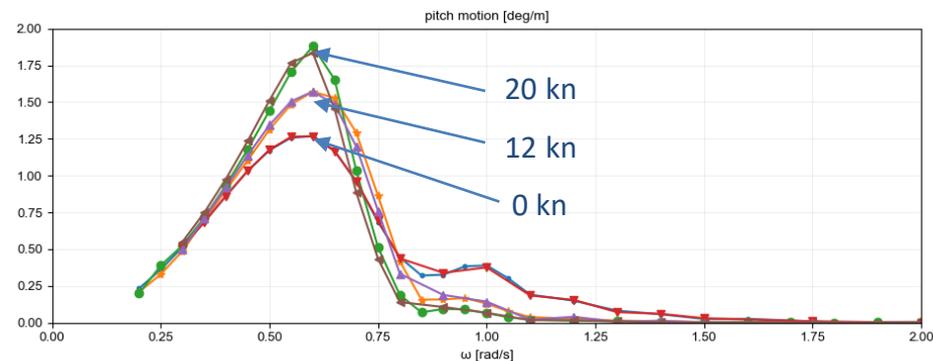
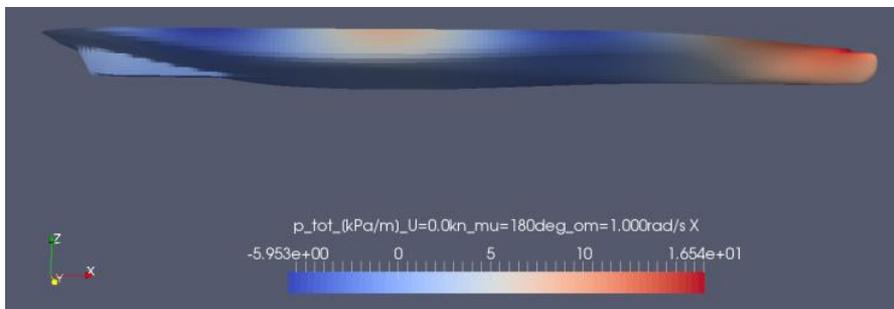
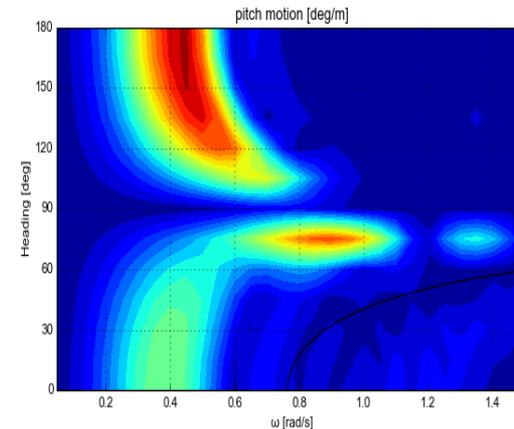
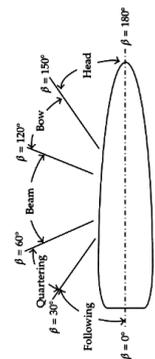
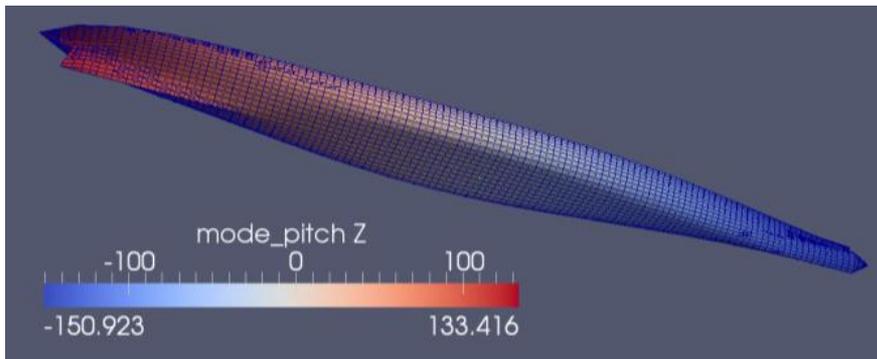
- Ship geometry described by **panels** (2k-15k)
- Excellent prediction at zero speed
- 5 speeds, 13 headings and 30 frequencies  
(=2000 **conditions**) -> 12h on 4 CPUs

## Rankine Source theory

- Ship and water FS described by **panels** (35k-85k)
- Excellent prediction at zero and forward speed
- 5 speeds, 13 headings and 30 frequencies  
(=2000 **conditions**) -> 8h on 200 CPUs

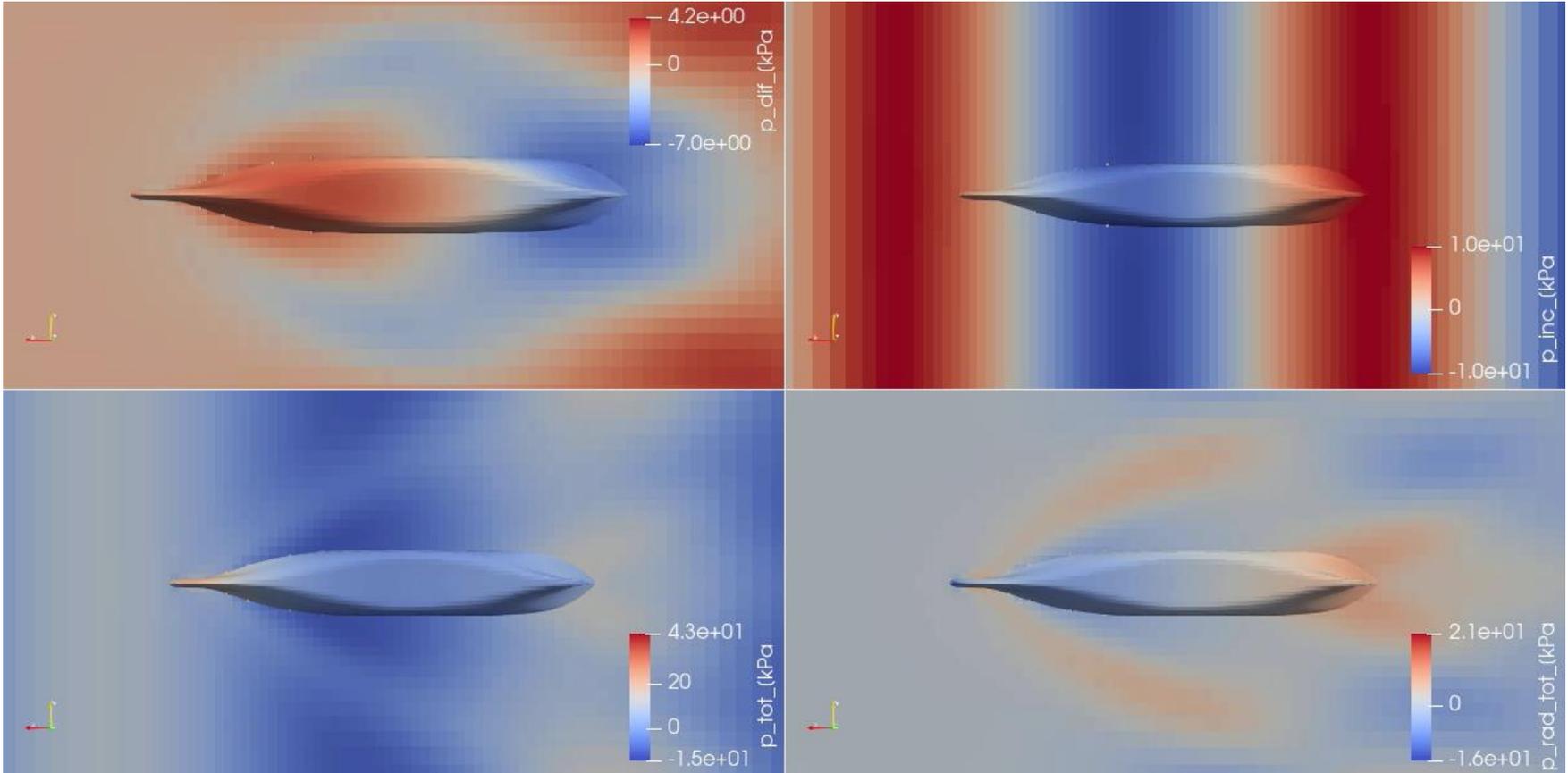
## Alternative method: CFD (non-linear effects)

- Excellent prediction at zero and forward speed
- 1 speeds, 1 heading and 1 frequency  
(=1 **condition**)-> 36h on 200 CPUs (x9.000 slower...)



SEAkeeping CALculations

Response Amplitude Operator (RAO)



Hydrostatics

Hydrodynamics

RankineSrc

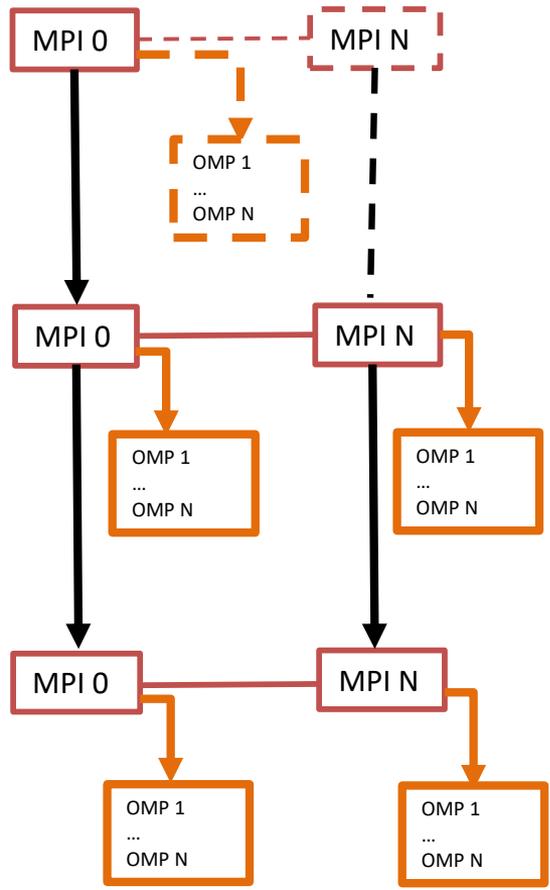
- AddNodes(MPI)
- Panels(MPI)
- GreenFunc(OMP)

RadDifSol

- Conditions(MPI)
- Solver(OMP)

Responses

DriftForces(MPI)  
TimeEvolution(OMP)



MPI: Loop Condition: U, mu, om, Aphi  
OMP: Loop Panel: Ship, Water FS

```
# i_cond, i_mu, i_U, i_om, mpiproc_cond(i_cond)
1 1 1 1 0
2 1 1 2 1
3 1 1 3 2
4 1 2 1 3
5 1 2 2 4
6 1 2 3 5
```

```
# mpi_size 6
#N_FreeSurf_panel,N_hydro_panel_total
25376 7048
#FSPanels_per_Rank,HydroPanels_per_Rank
4229 1174
#i_rank,N_FreeSurf_panel_rank_sta,N_FreeSurf_panel_rank_end
1 1 4229
2 4230 8458
3 8459 12687
4 12688 16916
5 16917 21145
6 21146 25376
#i_rank,N_hydro_panel_rank_sta,N_hydro_panel_rank_end
1 1 1174
2 1175 2348
3 2349 3522
4 3523 4696
5 4697 5870
6 5871 7048
```

## Cmake (software compilation process)

Compiler flags (platform/compiler/version dependent). External Libs. MSVS solution file (Win). Eclipse solution file (Linux). Installation step.

## Windows: Compiler: ifort (Intel); MPI: MS-MPI & IntelMPI

```
/NODEFAULTLIB #extLib compiler flags
/fpp /traceback /Qopenmp /Qmkl:parallel /libs:dll
set_source_files_properties(XXX PROPERTIES COMPILE_FLAGS/Qoverride-limits) #OMP compilation
/STACK:268435456,268435456 #OMP run-time
set_source_files_properties(YYY PROPERTIES HEADER_FILE_ONLY TRUE)
target_compile_options(${MARIN_MODULE} PUBLIC XXX/Include/mpi.f90) #MS-MPI Lib
target_link_libraries(${MARIN_MODULE} XXX/Lib/x64/msmpifec.lib) #MS-MPI Lib
target_link_libraries(${MARIN_MODULE} XXX/Lib/x64/msmpi.lib) #MS-MPI Lib

INSTALL(FILES <INTEL>/windows/mpi/intel64/bin/release/impi.dll"
  DESTINATION "${CMAKE_HOME_DIRECTORY}/eoe/bin/win64" RENAME msmpi.dll) #INTEL-MPI Lib
```

## Linux: mpiifort (Intel) & mpifort (GNU)

```
-fopenmp -fbacktrace -heap-arrays -ffree-line-length-512 -foverride-limits # mpifort (gnu)

Compiler optimizations #NO SIMD | NO AVX512; Only -O2
-----
ulimit -s unlimited #omp run-time
setenv MKL_NUM_THREADS 1
setenv OMP_NUM_THREADS 1
setenv OMP_STACKSIZE 250M #omp run-time
```

## MPI parallelization

```

#ifdef use_mpi_f08
#ifdef _WIN32
use mpi_f08
#endif
#endif

#ifdef use_mpi_f08
#ifdef _WIN32
include 'mpif.h'
#endif
#endif

```

Example: Loop over conditions

```

iCond MPI : do i_cond=1,N_U*N_omega_tot*N_mu ! loop over all input conditions

    if (mpiproc_cond(i_cond) .ne. my_rank) cycle iCond MPI ! skip if not my rank

    i_U = i_U_all(i_cond)      ! retrieve input indices
    i_om = i_om_all(i_cond)
    i_mu = i_mu_all(i_cond)

    U = thisSim%parHYD%shipSpeed(i_U)      ! use input indices
    mu = thisSim%waveSys%waveDir(i_mu)

module MPI_Calc
public :: &
    InitEndMPI,      &      ! start and end MPI interface
    prepareMPI,      &      ! distribute conditions over processes
    prepareMPIwAphi, &      ! distribute conditions, including roll input, over processes
    prepareMPI_panels, &      ! distribute panels over processes
    prepareMPI_AddNode, &      ! distribute added nodes processes
    get_MPIprocCond, &      ! retrieve process number on which a condition is calculated

```

MPI-3 shared memory (easy to implement!)

## OMP parallelization

Current approach: OMP Low level panel loop

```
!$OMP PARALLEL DEFAULT(SHARED) PRIVATE(i_panel)
!$OMP DO
  do i_panel=1,N
    call NumIntGreen_interface(&
      &A=A(i_panel,1:3), & ! unit = m,m,m
      &B=B(i_panel,1:3), & ! unit = m,m,m
      &C=C(i_panel,1:3), & ! unit = m,m,m
      &D=D(i_panel,1:3), & ! unit = m,m,m
      &P=P, & ! unit = m,m,m
      &k_e_inf_h=k_e_inf_h, & ! unit = -
      &k_e_inf=k_e_inf, & ! unit = -
      &lambda_e_inf=lambda_e_inf, & ! unit = m
      &area123=area123(i_panel), &
      &area341=area341(i_panel), &
      &area_max=area_max, & ! unit = m2
      &IFS=IFS, IBS=IBS, &
      &G=G(i_panel), Gx=Gx(i_panel), Gy=Gy(i_panel), Gz=Gz(i_panel))
  enddo
!$OMP END DO
!$OMP END PARALLEL

!$OMP PARALLEL DEFAULT(SHARED) PRIVATE(i_panel, intG)
!$OMP DO
  do i_panel=1,N
    call Rankine_singularity_G(intG, A(i_panel,1:3), B(i_panel,1:3),C(i_panel,1:3),
      D(i_panel,1:3), P, epsilon, this_analytical)

    intG_row(i_panel)=intG
  enddo
!$OMP END DO
!$OMP END PARALLEL
```

Previous approach: OMP high level input cond. loop

```
#ifdef highLevel_omp
!$omp parallel do schedule (dynamic), &
!$omp private(thisThread, i_cond, i_U, i_mu, i_om, U, mu, tau, &
!$omp phi_wl, pres_wl, prs_wl_db, prs_wl_uniform, prs, prs_db, &
!$omp prs_uniform, &
!$omp zeta, zeta_wl_db, zeta_wl_uniform, &
!$omp int_gradG2, &
!$omp j_count, j_count_start, i_count, i_count_start, singular, &
!$omp suppressIrregFreq, fid_hydfide, fn, &
!$omp mat_temp, i_mode, i_ship, i_panel, j_ship, j_panel, &
!$omp exists, stat, N_message, messages, sResult, error, error_marin, &
!$omp all_A, all_B, all_C, all_D, &
!$omp all_area123, all_area341, &
!$omp all_G, all_Gx, all_Gy, all_Gz, &
!$omp puvw, row_phi_uvw), &
!$omp firstprivate(tempRegWave, &
!$omp A_j, area_max, DD, h, k_e_inf, k_e_inf_h, KR, KZ, &
!$omp lambda_e_inf, omega_0, omega_e, &
!$omp RP, R, RSD, PVI, X, Y, Z, factor, &
!$omp P_add_wl, TERM, v_s, grad_h, P_add, &
!$omp v, v_inc, v_wl, &
!$omp grad_v, int_dG2dn_i, int_G2 , phi, D_R_hydro, &
!$omp rhsMat_dble, rhsMat_grad_dble, rhsMat_even_dble, &
!$omp rhsMat_even_grad_dble, rhsMat_odd_dble, rhsMat_odd_grad_dble, &
!$omp sol_even_dble, sol_even_grad_dble, sol_odd_dble, sol_odd_grad_dble, &
!$omp sol_GHRES_dble, sol_grad_dble, last_omega_e, ipiv, ipiv_even, ipiv_odd,
!$omp symIndex, ALL_sym, P_i, P_j, sol_GHRES_RANKINE, &
!$omp i_pn1, j_pn1)
#endif
  do i_cond=1,N_U*N_omega_tot*N_mu ! i_cond
```

integer :: N\_omega = HUGE\_R ! implied SAVE ; SUPER Dangerous!

Thread safe; OMP CRITICAL

Missing feature: OMP private copies of structure child's thisSim%A(i, j, k)

```
allocate(tempMesh_i(n_threads, N_ship))
! Create local copies for each omp thread
do i_threads = 1, n_threads
  do i_ship = 1, N_ship
    call Copy(thisMeshOrig=thisSim%ship(i_ship)%hydroMesh%mesh,
      thisMeshCopy=tempMesh_i(i_threads, i_ship))
  enddo
enddo
```



## Linux

(IntelMPI) \$>mpirun -n 16 SEACAL -c XXX

(OpenMPI) \$>mpirun --use-hwthread-cpus -n 16 SEACAL -c XXX

```

INFO: end of Sim_CalcRankineSrc <<<
This message was sent from Sim_CalcRankineSrc (logLevel=3), at 2021-05-21 09:25:14.476

INFO: Method steady flow AUTOMATIC selected
This message was sent from HYDCAL_R (logLevel=2), at 2021-05-21 09:25:14.497

INFO: Method wave modeling RANKINE selected
This message was sent from HYDCAL_R (logLevel=2), at 2021-05-21 09:25:14.501

INFO: Method steady flow is set to DOUBLE-BODY
This message was sent from HYDCAL_R (logLevel=2), at 2021-05-21 09:25:14.502

INFO: start of Sim_CalcSteadyFlowVel >>>
This message was sent from Sim_CalcSteadyFlowVel (logLevel=3), at 2021-05-21 09:25:14.502

INFO: end of Sim_CalcSteadyFlowVel <<<
This message was sent from Sim_CalcSteadyFlowVel (logLevel=3), at 2021-05-21 09:32:51.544

INFO: start of Sim_CalcRadDifSol >>>
This message was sent from Sim_CalcRadDifSol (logLevel=3), at 2021-05-21 09:33:00.965

INFO: Symmetry found
This message was sent from Sim_CalcRadDifSol (logLevel=3), at 2021-05-21 09:33:43.604

INFO: Size sysMat_even/odd = 0.028228 (GB)
This message was sent from Sim_CalcRadDifSol (logLevel=3), at 2021-05-21 09:34:25.833

INFO: Size D_R_hydro = 1.219015 (GB) ←
This message was sent from Sim_CalcRadDifSol (logLevel=3), at 2021-05-21 09:34:40.576
    
```

```

top - 09:35:15 up 3 days, 14:07, 4 users, load average: 19.55, 22.99, 24.43
Tasks: 633 total, 19 running, 614 sleeping, 0 stopped, 0 zombie
%Cpu(s): 99.7 us, 0.1 sy, 0.0 ni, 0.0 id, 0.0 wa, 0.2 hi, 0.0 si, 0.0 st
MiB Mem : 257732.5 total, 8483.3 free, 30363.3 used, 218886.0 buff/cache
MiB Swap: 1024.0 total, 1016.7 free, 7.3 used, 220787.4 avail Mem
    
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2565332	scostam+	20	0	7486000	1.5g	53380	R	100.0	0.6	10:25.31	SEACAL
2565353	scostam+	20	0	7481904	1.5g	53680	R	100.0	0.6	10:46.26	SEACAL
2565346	scostam+	20	0	7876228	3.0g	1.2g	R	99.7	1.2	10:21.94	SEACAL
2565347	scostam+	20	0	7486000	1.5g	53368	R	99.7	0.6	10:28.15	SEACAL
2565348	scostam+	20	0	7483952	1.5g	53372	R	99.7	0.6	10:21.07	SEACAL
2565349	scostam+	20	0	7481904	1.5g	53800	R	99.7	0.6	10:54.08	SEACAL
2565350	scostam+	20	0	7483952	1.5g	54132	R	99.0	0.6	10:30.06	SEACAL
2565351	scostam+	20	0	7481904	1.5g	53816	R	98.3	0.6	10:42.96	SEACAL
2565355	scostam+	20	0	7483952	1.5g	53804	R	98.3	0.6	10:54.74	SEACAL
2565357	scostam+	20	0	7486000	1.5g	53844	R	98.3	0.6	10:55.23	SEACAL
2565356	scostam+	20	0	7483952	1.5g	53540	R	97.4	0.6	10:46.32	SEACAL
2565358	scostam+	20	0	7486000	1.5g	53368	R	96.4	0.6	10:50.94	SEACAL
2565354	scostam+	20	0	7481908	1.5g	53740	R	95.7	0.6	10:46.56	SEACAL
2565359	scostam+	20	0	7486000	1.5g	53692	R	95.0	0.6	10:56.42	SEACAL
2565361	scostam+	20	0	7483952	1.5g	54032	R	89.4	0.6	11:02.71	SEACAL
2565360	scostam+	20	0	7486000	1.5g	53776	R	70.9	0.6	10:50.40	SEACAL
2576880	bamboo	20	0	666392	632396	31400	R	49.3	0.2	0:22.73	fortcom
346240	scostam+	0	-20	11.2g	527452	220208	S	3.6	0.2	19:08.79	nxnode,bin
346089	root	0	0	217008	41432	8764	S	2.3	0.0	56:20.60	sssd_kcm
346709	scostam+	20	0	3775748	354696	137804	S	2.3	0.1	12:19.55	gnome-shell
868341	scostam+	20	0	129752	16004	11084	R	0.3	0.0	3:00.91	top
2538004	scostam+	0	-20	1901548	103052	24636	S	0.3	0.0	0:16.12	nxnode,bin
1	root	20	0	253508	12052	8372	S	0.0	0.0	40:09.35	systemd

D\_R\_hydro Matrix => 1.2 GB MPI-3 shared memory ←

## Run-Time Memory Usage Monitor

### Windows

```
#Nr,PhysMTot(GB), PhysMAvail(GB), PhysMAvail(%)
#Sim_Init ONE INIT started
  1  34.131  21.779  64.000
#Sim_Init ONE INIT finished
  2  34.131  21.779  64.000
#SEACAL started
  3  34.131  21.779  64.000
...
#RESCAL finished
 10  34.131  20.803  61.000
#SEACAL finished
 11  34.131  20.898  62.000
```

<https://forums.developer.nvidia.com/t/how-to-use-the-function-globalmemorystatus-in-pg/132018>

### Linux

```
#Nr|mpirank|MemTot(GB)|memUsage(GB)|memPeak(GB)|FreeMem(GB)|FreeMem(%)
#SEACAL started
  1  10  377.3  4.3  4.3  355.4  94.2
#HYDMES finished
  2  10  377.3  4.3  4.3  354.8  94.0
...
#RESCAL finished
 176 10  377.3  6.8  7.4  305.4  80.9
#SEACAL finished
 177 10  377.3  6.4  7.4  314.2  83.3

!--- get process ID

pid=getpid()
write(pid_char,'(I8)') pid
filename='/proc/'//trim(adjustl(pid_char))//'/status'
```

- Low memory available: Hard to predict problems when big chunk (contiguous) memory allocation required
- Built-in functions for retrieving memory consumption info missing
- Adding extra elements to an array should be more easy... (other than via `move_alloc`)

## Temp files: file format

```

AddNode_hyd_rank<X>.h5           ; info for reference points stored per point
AddNodeWL_hyd_rank<X>.h5        ; info for water line points stored altogether
J-Class_FreeSurf_U_<>kn_mu_<>deg_om0_<>.hyd ; info for a given condition for FS panels
J-Class_HydroMesh_U_<>kn_mu_<>deg_om0_<>.hyd ; info for a given condition for Hydro Mesh
J-Class_Sigma_U_<>kn_mu_<>deg_om0_<>.hyd   ; info of Sigma matrix for a given condition

```

.hyd (fort binary) file per condition (U, mu, om)  
 (allowing “re-start” with adding extra cond’s)

```

J-Class_FreeSurf_U_0.0kn_mu_225deg_om0_0.300.hyd
J-Class_FreeSurf_U_0.0kn_mu_225deg_om0_0.500.hyd
J-Class_FreeSurf_U_0.0kn_mu_225deg_om0_0.700.hyd
J-Class_FreeSurf_U_0.0kn_mu_225deg_om0_0.900.hyd
J-Class_FreeSurf_U_0.0kn_mu_225deg_om0_1.100.hyd
J-Class_FreeSurf_U_0.0kn_mu_45deg_om0_0.300.hyd
J-Class_FreeSurf_U_0.0kn_mu_45deg_om0_0.500.hyd
J-Class_FreeSurf_U_0.0kn_mu_45deg_om0_0.700.hyd
J-Class_FreeSurf_U_0.0kn_mu_45deg_om0_0.900.hyd
J-Class_FreeSurf_U_0.0kn_mu_45deg_om0_1.100.hyd

```

Switching to hdf5  
 (per MPI rank)



```

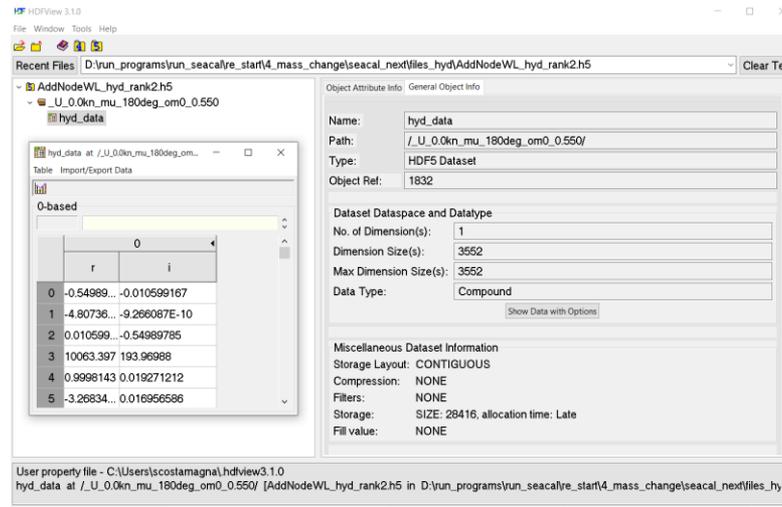
AddNodeWL_hyd_rank0.h5
AddNodeWL_hyd_rank10.h5
AddNodeWL_hyd_rank11.h5
AddNodeWL_hyd_rank12.h5
AddNodeWL_hyd_rank13.h5
AddNodeWL_hyd_rank14.h5
AddNodeWL_hyd_rank1.h5
AddNodeWL_hyd_rank2.h5
AddNodeWL_hyd_rank3.h5
AddNodeWL_hyd_rank4.h5
AddNodeWL_hyd_rank5.h5
AddNodeWL_hyd_rank6.h5
AddNodeWL_hyd_rank7.h5
AddNodeWL_hyd_rank8.h5
AddNodeWL_hyd_rank9.h5

```

Too many files (>2000)

Too much disk space (>10Gb)

```
AddNodeWL_hyd_rank0.h5  
AddNodeWL_hyd_rank10.h5  
AddNodeWL_hyd_rank11.h5  
AddNodeWL_hyd_rank12.h5  
AddNodeWL_hyd_rank13.h5  
AddNodeWL_hyd_rank14.h5  
AddNodeWL_hyd_rank1.h5  
AddNodeWL_hyd_rank2.h5  
AddNodeWL_hyd_rank3.h5  
AddNodeWL_hyd_rank4.h5  
AddNodeWL_hyd_rank5.h5  
AddNodeWL_hyd_rank6.h5  
AddNodeWL_hyd_rank7.h5  
AddNodeWL_hyd_rank8.h5  
AddNodeWL_hyd_rank9.h5
```



The screenshot shows the HDFView 3.1.0 interface. The main window displays a tree view of the file structure, with the dataset 'hyd\_data' selected. A table window is open, showing the data for 'hyd\_data' at the path '/\_U\_0.0kn\_mu\_180deg\_om0\_0.550/'. The table has two columns, 'r' and 'i', and six rows of data.

	r	i
0	-0.54989...	-0.010599167
1	-4.80736...	-9.266087E-10
2	0.010599...	-0.54989785
3	10063.397	193.96988
4	0.9998143	0.019271212
5	-3.26834...	0.016956586

The right-hand pane shows the 'Object Attribute Info' for the selected dataset. The 'Name' is 'hyd\_data', the 'Path' is '/\_U\_0.0kn\_mu\_180deg\_om0\_0.550/', the 'Type' is 'HDF5 Dataset', and the 'Object Ref' is '1832'. The 'Dataset Dataspace and Datatype' section shows 'No. of Dimension(s): 1', 'Dimension Size(s): 3552', and 'Max Dimension Size(s): 3552'. The 'Data Type' is 'Compound'. The 'Miscellaneous Dataset Information' section shows 'Storage Layout: CONTIGUOUS', 'Compression: NONE', 'Filters: NONE', 'Storage: SIZE: 28416, allocation time: Late', and 'Fill value: NONE'.

-H5 (hdf5) files with *groups* per (U, mu, om) condition

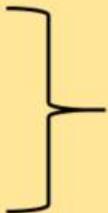
**PROBLEM to deal with...** : opening/closing h5 files repeatedly  
time consuming, fragmented file resulting in huge file size (x20)

-hdf5 lib compiled with FORTRAN2003 flag ON

-H5\_INTERFACE Module file

## Parallel settings selection | SEACAL inputs

- i) Nr. Total MPI
- ii) MPI x Node
- iii) openMP x Node



Based on

- Nr. input conditions
- ( $\mu \cdot \omega \cdot U \cdot A_{phi}$ )
- Nr. Panels
- ( Ship + FreeSurf (Rankine) )
- HARDWARE

Total nr. of panels (Hydro+FreeSurf)	Estimated memory consumption (Gb)	queue = LARGE NrPipesNot	queue = Normal, Short NrPipesNot
5.000	0.545571918	24	24
10.000	1.1370817	24	24
15.000	1.878433825	24	24
20.000	2.7683268	24	24
25.000	3.806760625	24	24
30.000	4.9937353	24	18
35.000	6.329250825	24	14
40.000	7.8133072	24	12
50.000	11.2270425	24	8
60.000	15.2349412	24	6
80.000	25.0332288	15	3
100.000	37.20817	10	2
120.000	51.7597648	7	1
140.000	68.6880132	6	1
160.000	87.9929152	4	1
200.000	133.73268	2	-
250.000	204.2760625	1	-
300.000	289.67353	1	-
340.000	368.6864452	1	-

