



Computational real time simulation, decomposition of mathematical models with algorithms of interactive control and visualization

Alexander Degtyarev, Vasily Khramushin

a.degtyarev@spbu.ru, khram@mail.ru

Saint Petersburg State University

Direct simulation of physical phenomena



Points for direct simulation.

1. Increased computing power
2. Possibility of a comprehensive description of the phenomenon, if its physics is known; application of fundamental conservation laws
3. Possibility of virtual testbed organization – virtual computational environment where natural processes, external excitations, technical objects are simulated
4. Possibility to provide wide possibilities of modeling: object parameters, external conditions, extreme scenarios, etc.

Direct simulation of physical phenomena



Merits of direct simulation.

1. No scale effect in model experiments
2. Accurate knowledge of environment state and characteristics of discovered objects
3. Any experimental conditions, including extreme ones, or those that cannot be reproduced under the conditions of a model experiment
4. Significant difference in the cost of organizing and conducting an experiment

Direct simulation of physical phenomena



Problems of direct simulation.

1. Direct simulation is not just computation
2. As any experiment, it imposes
 1. Planning
 2. Control
 3. Interaction
 4. Recording
3. Additional needs to models and algorithms

Interactive control and visualization of the direct simulation process

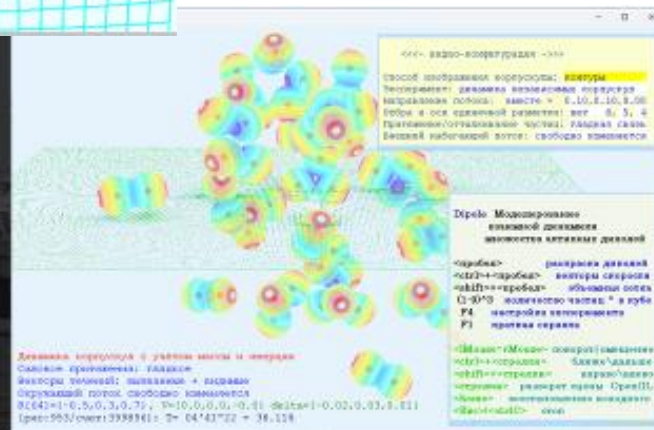
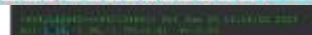
Components of direct simulation



1. Solution decomposition by physical processes in mathematical representation
2. Multi-window graphical visualization toolkit independent of the computing environment
3. Time synchronization system with interactive control of computational model parameters

1. *Mathematical modeling itself*
2. *Automatic control*
3. *Graphical visualization of results*

Purely nonlinear wave and corpuscular models of mechanics



Integrated models of ship hydromechanics in storm maneuvering

Outline



Defining the range of tasks to be solved.

1. Designing and staging a direct computational simulation
 - 1.1. *Differential models or finite volume methods*
 - 1.2. *Small spatial differences and time increments*
 - 1.3. *Non-stationary spatial phenomena in time domain*
2. Decomposition of computational processes and computer architecture
 - 2.1. *Establishing resource-intensive nonstationary processes*
 - 2.2. *Real-time applications or engineering tasks*
3. Examples and peculiarities of research and application problems
 - 3.1. *Corpuscular mechanics and storm hydromechanics of a ship*

Software tools 1



Algorithmic problem statement

Two variants of direct simulation.

- A. Resource-intensive computing process is optimized only for maximum efficiency and computation speed.
 - a.1. The number of graphic and control procedures, which can be suspended as needed, is reduced.*
- B. Direct computational simulation with realization of complex physical and mathematical models in real time.
 - b.1. Numerical arrays are optimized to describe the simulated objects and the surrounding continuous environment.*
 - b.2. Interactive control options are expanding.*

Software tools 2



Decomposition of computational processes by execution time

Given the multi-threaded execution of mathematical models themselves, breaking the computational process is fundamentally impossible, however, each clock cycle of a computational simulation can be limited to a relatively small quantum of time.

1. *Variant A – graphical operations with interactive control are performed at moments of suspension of execution of key mathematical models.*
2. *Variant B – all operations are executed in parallel by using independent interval timers in parallel computing threads.*

Software tools 3



Variant A – single flow of execution, control and visualization.

Separation of procedures in successive quanta of time

```
long WaitTime(  
    long wait,                // delay for independent interrupt processing  
    bool( *inFree )() = null, // free function of the computational experiment  
    long work = 0 );          // control time to execute the calculation cycle [ $\mu$ s]
```

Examples:

WaitTime(wait) – Sleep function analog (μ s) with processing of interactive requests

WaitTime(wait, inFree) – math model call inFree with time interval μ s

WaitTime(wait, inFree, work) – cyclic startup with time-controlled execution with pause for control queries and presentation of results.

WaitTime – creates a continuous cycle of the computational experiment, time intervals can be reconfigured by exiting the cycle InFree() \neq false.

Software tools 4



Variant A – time samples to optimize and accelerate calculations

```
long StartTime,           // computer start time of the whole program
    RealTime,             // current execution time of the inFree procedure within WaitTime
    GetTime(),            // query the exact time in milliseconds (GetTickCount)
    ElapsedTime();       // querying the program runtime( $\mu$ sec)
```

Depending on the performance of the computer's CPU, to create comfortable conditions for interactive control and obtain a relatively smooth frame-by-frame sweep without affecting the computational experiment, it is necessary to dynamically adjust the parameters of the control procedure **WaitTime**(wait,inFree(),work).

Software tools 5



Variant A – interactive experiment control queries

```
byte Window::WaitKey()    // stop and waiting of new symbol from keyboard
byte Window::GetKey()     // querying and selecting a symbol without stopping the program
byte Window::ScanKey()    // symbol polling without stopping and without sampling from the
                           queue
byte Window::ScanStatus() // getting associated keys code from buffer
```

Instead of the usual system prompts to the keyboard, it is possible to bind the main loop of interactive requests to a specific graphical window.

Thus, the simplest version of the software environment for the computational experiment is implemented. It is possible to dynamically adjust the speed of calculations.

Software tools 6



Variant B – parallel threads for mathematical modeling, graphical visualization and interactive control.

1. The real-time experiment must initially corresponds the performance of the computer in terms of the amount of computation.
2. Taking into account the need to work in an environment with many heterogeneous interrupts, it is possible to use object-oriented programming methods using virtual procedures to speed up development..
3. Interrupt handling is partly formalized, however, in OOP, it is advisable for the developer to have an open Window-Place code for choosing procedural "inheritances", or modifying it.

Software tools 7



Variant B – interval timer in OpenGL environment.

```
virtual Window& Window::Timer()           // virtual procedure for timer processing
    Window& Window::SetTimer( μsec, bool( *inFree )()=null ) // interval and transaction
    Window& Window::KillTimer()             // timer reset
```

The use of the interrupt tool when calling free transactions inFree (), and in reverse sequences of calls over **Virtual**, allow you to automate the recovery of the graphical context of OpenGL.

1) Using virtual multiplies the program's executable code due to the automatic involvement of a huge library LibStd++

Software tools 8



Variant B – similar processing of interrupts from external devices.

```
virtual Place& Place::Draw()           // virtual procedure of image update
virtual Place& Place::Mouse( x,y )     // motion in the field of graphical area
virtual Place& Place::Mouse( state, x,y ) // reaction when the cursor keys are pressed
virtual bool Window::KeyBoard( byte )   // virtual procedure input from keyboard

Place& Place::Draw( bool(*inDraw)() )   // reference to the external rendering process
Place& Place::Mouse( bool(*inPass)( int,int ) ) // external processing
Place& Place::Mouse( bool(*inPush)( int,int,int ) ) // interrupt from mouse
Window& Window::KeyBoard( bool( *inKey )( byte ) ) // registration of a free interrupt
handling module to respond to keyboard input of commands or data.
```

Derived class Place – rectangular fragment inside the Window

Contraction of computational simulation



- 1 – Continuum-corpuscle kinematics;
- 2 – Wave dynamics;
- 3 ~ Ship hydrodynamics in heavy sea.

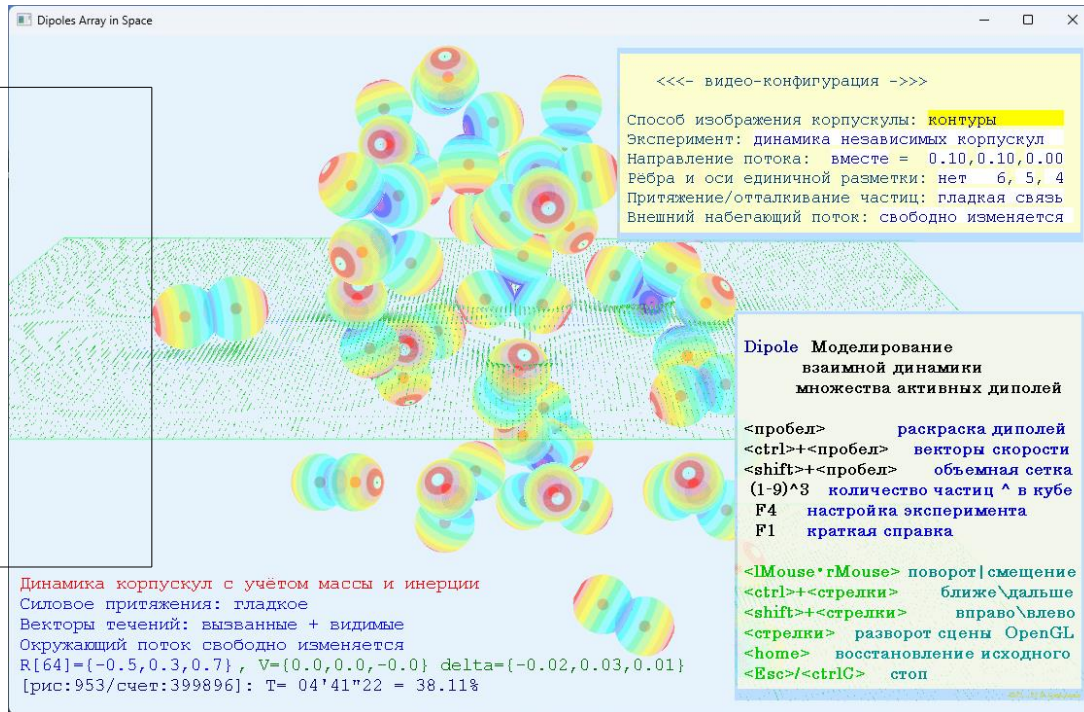
Algorithms variants :

- 1 – research of 3D models of tensor mathematics;*
- 2 + algorithms of nonstationary and nonlinear wave dynamics;*
- 3 = synthesis of numerical models in ship hydromechanics in heavy sea.*

Continuum-corpuscle kinematics

1 – research on numerical models of three-dimensional tensor mathematics

```
C:\Users\Khrum\VasilyKhrum > + <v>
columnsort
<624/142640>=<64/134651> Fri Jun 23 15:16:52 2023
R=(-2.14,-1.96,-1.79)=3.41 V=-0.00
R=( 0.13,-2.30,-2.55)=3.44 V=-0.00
R=( 0.11,-2.60,-3.68)=4.51 V=+0.00
R=(-0.24,-2.54,-0.95)=2.72 V=-0.00
R=(-2.68,-3.01,-2.04)=4.52 V=-0.00
R=(-1.04,-3.54,-0.43)=3.71 V=+0.00
R=( 0.66,-3.37,-0.63)=3.49 V=+0.00
R=(-2.87,-2.65,-1.38)=3.81 V=-0.02
R=(-2.51,-2.73,-0.90)=3.82 V=+0.00
R=(-0.69,-3.20, 0.65)=3.34 V=-0.00
R=( 0.68,-3.19, 0.52)=3.31 V=+0.00
R=( 2.06,-2.18,-0.60)=3.05 V=+0.02
R=(-2.37,-2.31, 3.50)=4.82 V=+0.00
R=(-1.41,-2.55, 3.57)=4.61 V=-0.00
R=( 0.71,-1.07, 0.19)=1.30 V=+0.00
R=( 2.50,-1.14, 2.31)=3.59 V=+0.00
R=(-2.26, 0.77,-2.59)=3.52 V=+0.00
R=(-0.54,-0.38,-3.30)=3.37 V=-0.00
R=( 0.70,-1.65,-3.79)=4.19 V=-0.00
R=( 2.27,-0.96,-2.53)=3.53 V=-0.00
R=(-4.03, 0.54, 1.72)=4.41 V=-0.00
R=( 0.03,-0.36, 0.71)=0.79 V=-0.00
R=( 1.44,-1.70,-2.43)=3.29 V=+0.01
R=( 3.38,-0.61,-0.58)=3.48 V=-0.00
R=(-3.07, 0.48, 0.79)=3.20 V=+0.00
R=(-1.50,-0.66, 2.55)=3.03 V=-0.00
R=( 3.68, 0.10, 2.77)=4.61 V=+0.00
R=( 3.14,-0.56, 0.58)=3.24 V=+0.00
R=(-2.35,-0.34, 3.29)=4.06 V=+0.00
R=(-0.40, 0.42, 3.82)=3.86 V=+0.00
R=( 0.96, 0.44, 3.81)=3.95 V=-0.00
R=( 3.12,-1.13, 3.21)=4.62 V=-0.00
```



Algorithms of nonlinear wave hydrodynamics



плоской прогрессивной трохидальной волны

Progressive – wind waves

Extremely high waves

Extremely standing waves

$$\frac{\rho \cdot V^2}{2} + \rho \cdot g \cdot \zeta_w = Const$$

Bernoulli's law for pressure and velocity in a stream of water flow

Computational simulation development

~ synthesis of numerical models in ship hydromechanics in heavy sea.

The text console is used, and only in it you can output information from the independent block of execution of mathematical models.

```
Штормовая мореходность корабля (вычислительный эксперимент)
Штормовая мореходность корабля (вычислительный эксперимент)
Ленинград, Кораблестроительный институт \ Санкт-Петербург, Государственный университет
Научно-инженерное общество судостроителей им.А.Н.Крылова,
подсекция мореходных качеств корабля в штормовых условиях
«МИДВ» ©1975-2023 Калининград-Севастополь-Ленинград-Сахалин - 17 17 17 - В.Храмушин

>>> Г-5 глиссер А.Н.Туполев
>>> { L=17.28, B=3.33, T=0.68, Psi= 00'\delta=0cm }^0.68 K(a.62<шп[60]>66.ф) Statum{ 4 }

Волна: L= 25 м, T= 4.0", A=0.60 м, C= 6.2 м/с, Dir=191°, Ds=3.1 м [68.46] = {207.138} м
Зыбь: 40 м, 5.1", 0.48 м, 7.9 м/с, 158°, 4.0 м [54.36]
Вал: 60 м, 6.2", 0.36 м, 9.7 м/с, 230°, 4.8 м [44.30]

Time 10'59" + 0.50" C:{ -0.9,-0.0,-0.3 } mC:| 0.69 -0.72 -0.00 | X 0°17
Speed 19.9уз(0.79=3.9L) W:{ -1.6,-0.0,-0.1 } | 0.72 0.69 -0.00 | Y -0°37
Volume 15 << 10 vW:{ 0.0,-0.0,1.0 } | 0.00 -0.00 1.00 | Z 46°37
Surface 54 << 48 mW:| 27 -44.2 |
Floatable 44 << 41 | -44.2 686 |
pCenter 1.7 >> 1.5 -- 0.5 Gravity.z
h 1.20 >> 0.95 -- pM 55.5 >> 44.4 inMass:| 10 -0.0 2.1 |
| -0.0 233 0.0 |
| 2.1 0.0 240 |
A Курс( δ )заданный => 46.0°<0.0> 45.0°
крен 0.7°, дифферент -1.8° vS:{ 0.1,0.1,14.1 }
Скорость 19.9 из 20.0 узлов vR:{ -0.1,2.6,0.9 }

C [м] - положение мгновенного центра величины (подводного объема)
mC (δ) - матрица ориентации корпуса (корабельные оси в морском базисе)
W [м] - корабельные координаты центра площади действующей ватерлинии
vW (δ) - интегральная нормаль к действию гидростатических сил всплытия
vS [Н/ρ] - накопительный вектор сил от перепадов уровня моря
vR [Н·м/ρ] - поворотный момент от криво-наклонной ватерлинии
mW [м□]4 - моменты инерции площади действующей ватерлинии
mM [м□]5 - объёмные моменты инерции погруженного корпуса
inMass - исходные моменты инерции корпуса
```

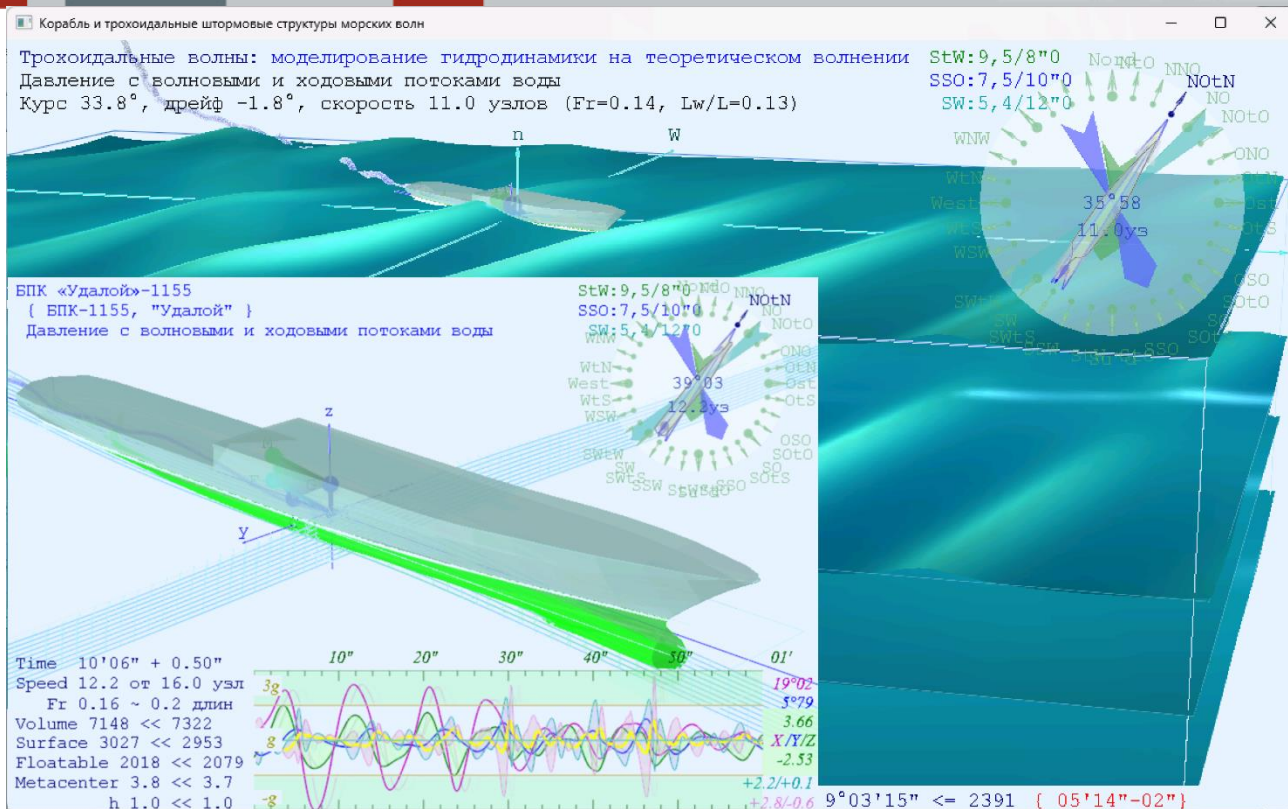


On the graphs: heaving, rolling and pitching, inside the hull constant and variable centers, stability parameters...

Computational simulation development 3

Group structure of intensive trochoidal waves.

The second graphical window shows ship behavior in heavy sea. It permits to maneuver between wave crests with different speeds and courses





Conclusion

1. The minimum necessary tools for controlling a direct computational simulation are presented
2. The requirements for designing a direct computing simulation based on the architecture of high-performance computing system are formulated
3. Versions of direct and virtual interactive interfaces based on parallel architecture and OpenGL graphics environment are proposed
4. Examples of application of new tools for problem solution environment (PSE) development are presented

Thanks for attention

