

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ПРИ СОДРУЖЕСТВЕ НГТУ И НГУ



**НГТУ
НЭТИ**

N* Новосибирский
государственный
университет
***НАСТОЯЩАЯ НАУКА**

Крупская Вероника Александровна, ПМ-83, +7996-543-26-21

Малышкин Виктор Эммануилович, д.т.н., профессор

*ОПТИМИЗАЦИЯ АЛГОРИТМОВ МОДУЛЯ
LGSEP CAD-СИСТЕМЫ BRICSCAD*

Новосибирск 2022

Цель работы и задачи

Цель

Ускорить решение задачи параметрического моделирования в модуле LGSEP CAD-системы BricsCad.

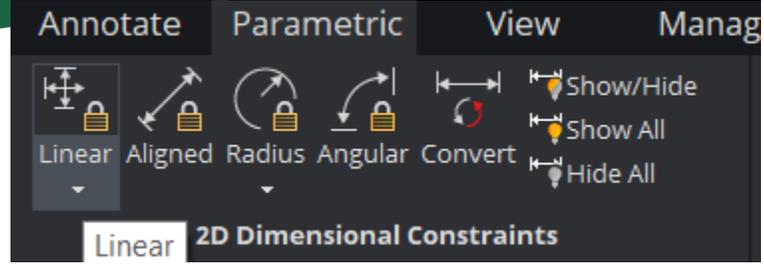
Задачи

- 1) Исследование предметной области;
- 2) Анализ существующего кода;
- 3) Программная реализация оптимизаций;
- 4) Экспериментальное исследование работы программной реализации;

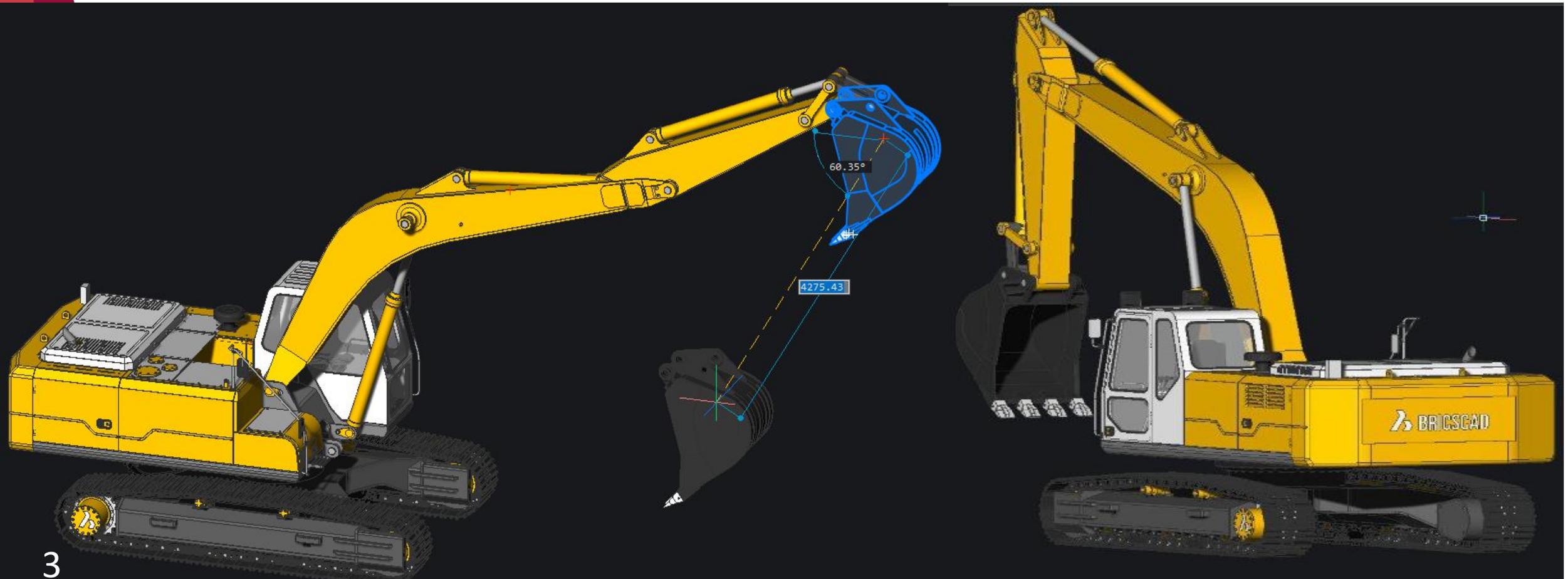


Введение

До



После



Жизненный цикл



Проектирование

Профилирование

NDA

	Total CPU [unit,...		
	24469 (98,99%)		
	24458 (98,94%)		
	24435 (98,85%)		143478 (99,05%)
	24429 (98,83%)		143444 (99,03%)
	24424 (98,81%)		143414 (99,01%)
	24424 (98,81%)		143413 (99,01%)
	22744 (92,01%)		143404 (99,00%)
	21072 (85,25%)		143403 (99,00%)
	9232 (37,35%)		143400 (99,00%)
	8786 (35,54%)		143315 (98,94%)
	5467 (22,12%)		143313 (98,94%)
	3255 (13,17%)		143310 (98,94%)
	3229 (13,06%)		143308 (98,94%)
	3072 (12,43%)		140957 (97,31%)
	2572 (10,40%)		121801 (84,09%)
	2533 (10,25%)		51307 (35,42%)
▶ LGSEPVectorCopy	2373 (9,60%)		
▶ CVectorDouble::Allocate	2321 (9,39%)		
▶ LGS3DMUCDifferentialNew::scalarProduct2			34159 (23,58%)

Опции

КОМПИЛЯЦИИ

O2(базовая)	15c
O1	17c
/arch:AVX2	16c
ffast-math	15c

Поиск уже существующих

- `#pragma omp` библиотеки OpenMP
- `std::thread` стандартной библиотеки языка C++
- `tbb` из библиотеки intel threading building blocks
- системная параллелизация `pthread`
- `CreateThread` из `winapi`

Векторизация?

```
Viewing Options
const std::vector<double>& vec2,
double iTolerancesRelation2 /*(linTolerance/angTolerance)^2*/) const
{
00007FF707DF44E0 sub     rsp,8
    double ret = 0;
    for (unsigned i = 0; i < _actualVarsIdxs.size(); i++)
00007FF707DF44E4 mov     r11,qword ptr [rcx+60h]
00007FF707DF44E8 xor     r9d,r9d
00007FF707DF44EB mov     r10,qword ptr [rcx+68h]
00007FF707DF44EF xorps  xmm1,xmm1
00007FF707DF44F2 sub     r10,r11
00007FF707DF44F5 sar     r10,2
00007FF707DF44F9 test   r10,r10
00007FF707DF44FC je      LGS3DMUCDifferentialNew::scalarProduct2+87h (07FF707DF4567h)
    {
        double a = vec1[i] * vec2[i];
00007FF707DF44FE mov     r8,qword ptr [r8]
00007FF707DF4501 mov     eax,r9d     $1ms elapsed
00007FF707DF4504 mov     qword ptr [rsp+10h],rbx
00007FF707DF4509 mov     rbx,qword ptr [rdx]
00007FF707DF450C mov     qword ptr [rsp],rdi
        ret += (!_isRotationalVariable[_actualVarsIdxs[i]]) ? a : a * iTolerancesRelation2;
00007FF707DF4510 mov     rdi,qword ptr [rcx+40h]
00007FF707DF4514 nop     dword ptr [rax]
00007FF707DF4518 nop     dword ptr [rax+rax]
    }
    {
        double a = vec1[i] * vec2[i];
00007FF707DF4520 movsd  xmm0,mword ptr [r8+rax*8]
        ret += (!_isRotationalVariable[_actualVarsIdxs[i]]) ? a : a * iTolerancesRelation2;
00007FF707DF4526 mov     edx,dword ptr [r11+rax*4]
00007FF707DF452A mulsd  xmm0,mword ptr [rbx+rax*8]
00007FF707DF452F movzx  eax,d1

```

Векторизация

```
#pragma omp simd  
_restrict
```

```
double tmp[4] = {};  
#pragma omp simd  
for (unsigned i = 0; i < _actualVarsIdxs.size() / 4; i++)  
{  
    double a[4];  
    a[0] = vec1[i * 4 + 0] * vec2[i * 4 + 0];  
    a[1] = vec1[i * 4 + 1] * vec2[i * 4 + 1];  
    a[2] = vec1[i * 4 + 2] * vec2[i * 4 + 2];  
    a[3] = vec1[i * 4 + 3] * vec2[i * 4 + 3];  
    double mult[4];  
    mult[0] = !_isRotationalVariable[_actualVarsIdxs[i * 4 + 0]] ? iTolerancesRelation2: 1.0;  
    mult[1] = !_isRotationalVariable[_actualVarsIdxs[i * 4 + 1]] ? iTolerancesRelation2: 1.0;  
    mult[2] = !_isRotationalVariable[_actualVarsIdxs[i * 4 + 2]] ? iTolerancesRelation2: 1.0;  
    mult[3] = !_isRotationalVariable[_actualVarsIdxs[i * 4 + 3]] ? iTolerancesRelation2: 1.0;  
    tmp[0] += a[0] * mult[0];  
    tmp[1] += a[1] * mult[1];  
    tmp[2] += a[2] * mult[2];  
    tmp[3] += a[3] * mult[3];  
}  
ret += tmp[0] + tmp[1] + tmp[2] + tmp[3];  
for (unsigned i = (_actualVarsIdxs.size() / 4) * 4; i < _actualVarsIdxs.size(); i++)  
{  
    const double a = vec1[i] * vec2[i];  
    ret += (!_isRotationalVariable[_actualVarsIdxs[i]]) ? a : a * iTolerancesRelation2;  
}
```

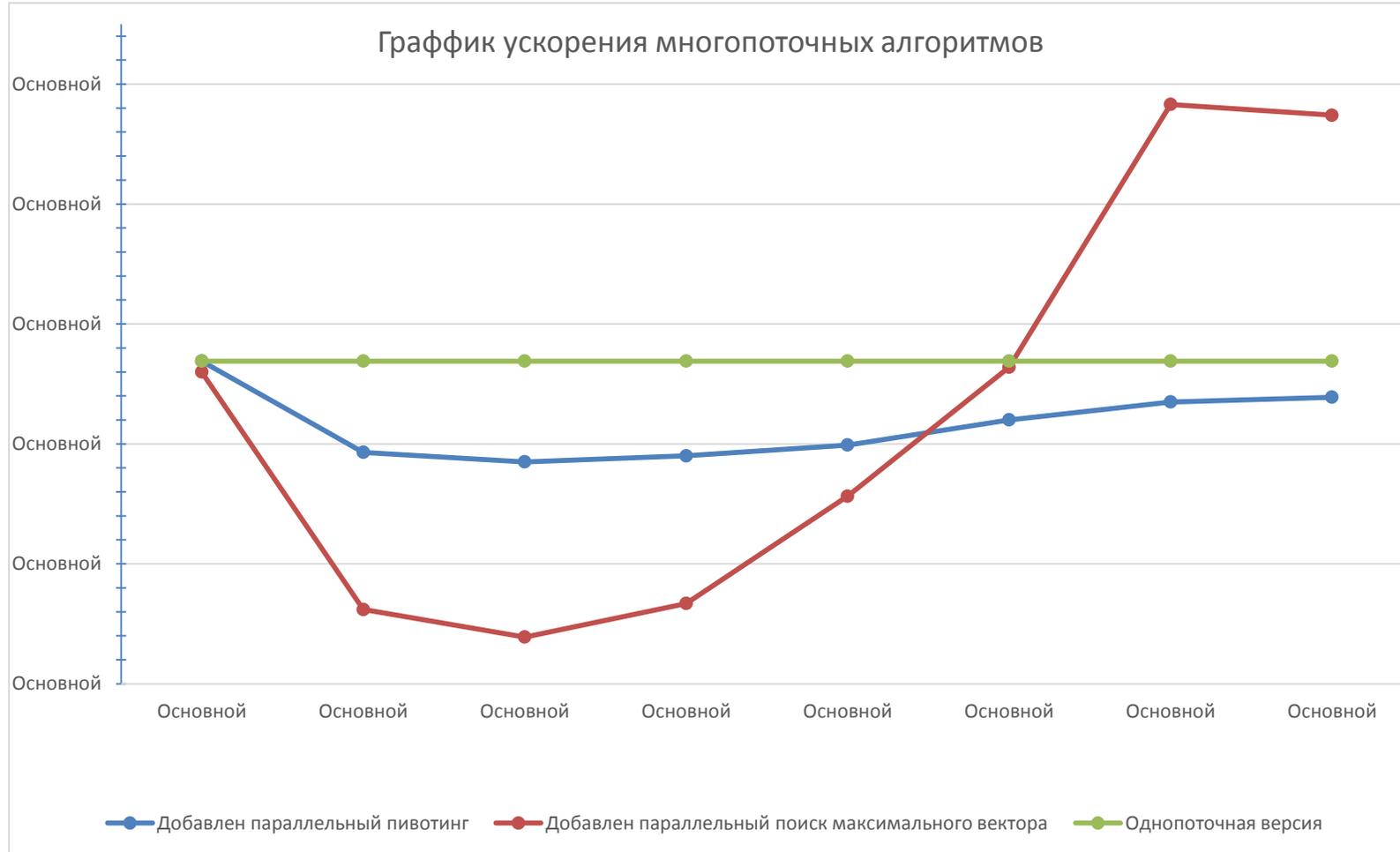
Векторизация

```
__m256d partial = {};  
for (unsigned i = 0; i < _actualVarsIdxs.size() / 4; i++)  
{  
    const __m256d v1 = _mm256_loadu_pd(vec1.data() + i * 4);  
    const __m256d v2 = _mm256_loadu_pd(vec2.data() + i * 4);  
    const __m256d dot = _mm256_mul_pd(v1, v2);  
    double mult[4] = {1.0, 1.0, 1.0, 1.0};  
    if (_isRotationalVariable[_actualVarsIdxs[i * 4 + 0]])  
        mult[0] = iTolerancesRelation2;  
    if (_isRotationalVariable[_actualVarsIdxs[i * 4 + 1]])  
        mult[1] = iTolerancesRelation2;  
    if (_isRotationalVariable[_actualVarsIdxs[i * 4 + 2]])  
        mult[2] = iTolerancesRelation2;  
    if (_isRotationalVariable[_actualVarsIdxs[i * 4 + 3]])  
        mult[3] = iTolerancesRelation2;  
    const __m256d vm = _mm256_loadu_pd(mult);  
    partial = _mm256_fmadd_pd(dot, vm, partial);  
}  
double tmp[4];  
_mm256_storeu_pd(tmp, partial);  
ret += tmp[0] + tmp[1] + tmp[2] + tmp[3];  
for (unsigned i = (_actualVarsIdxs.size() / 4) * 4; i < _actualVarsIdxs.size(); i++)  
{  
    const double a = vec1[i] * vec2[i];  
    ret += (!_isRotationalVariable[_actualVarsIdxs[i]]) ? a : a * iTolerancesRelation2;  
}
```

Арена памяти

Улучшенные структуры
данных

Метсру -> цикл





- ✓ Выявлены функции модуля LGSEP CAD-системы BricsCAD, которые занимают основное время решения задач параметрического моделирования, эти функции подлежат оптимизации в первую очередь;
- ✓ Разработаны варианты кода модуля LSPEG, реализующие ряд оптимизаций, в числе которых – реорганизация структуры данных для хранения и работы с матрицами, векторизация циклов, распараллеливание по потокам, замена динамического выделения памяти на статическое, ускорение с помощью использования ассемблерных специфичных для данной архитектуры инструкций;

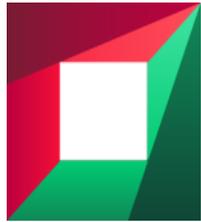
На ряде тестов показано ускорение решения задач параметрического моделирования до 20%, потенциально, на моделях большего размера

Направления дальнейшей работы:



Результаты соответствуют цели, хотя задержка отклика все еще является заметной для пользователя. Для дальнейшего ускорения планируется изучить и внедрить оптимизации под графические ускорители, являющиеся частью большинства клиентских систем ЭВМ.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ПРИ СОДРУЖЕСТВЕ НГТУ И НГУ



**НГТУ
НЭТИ**

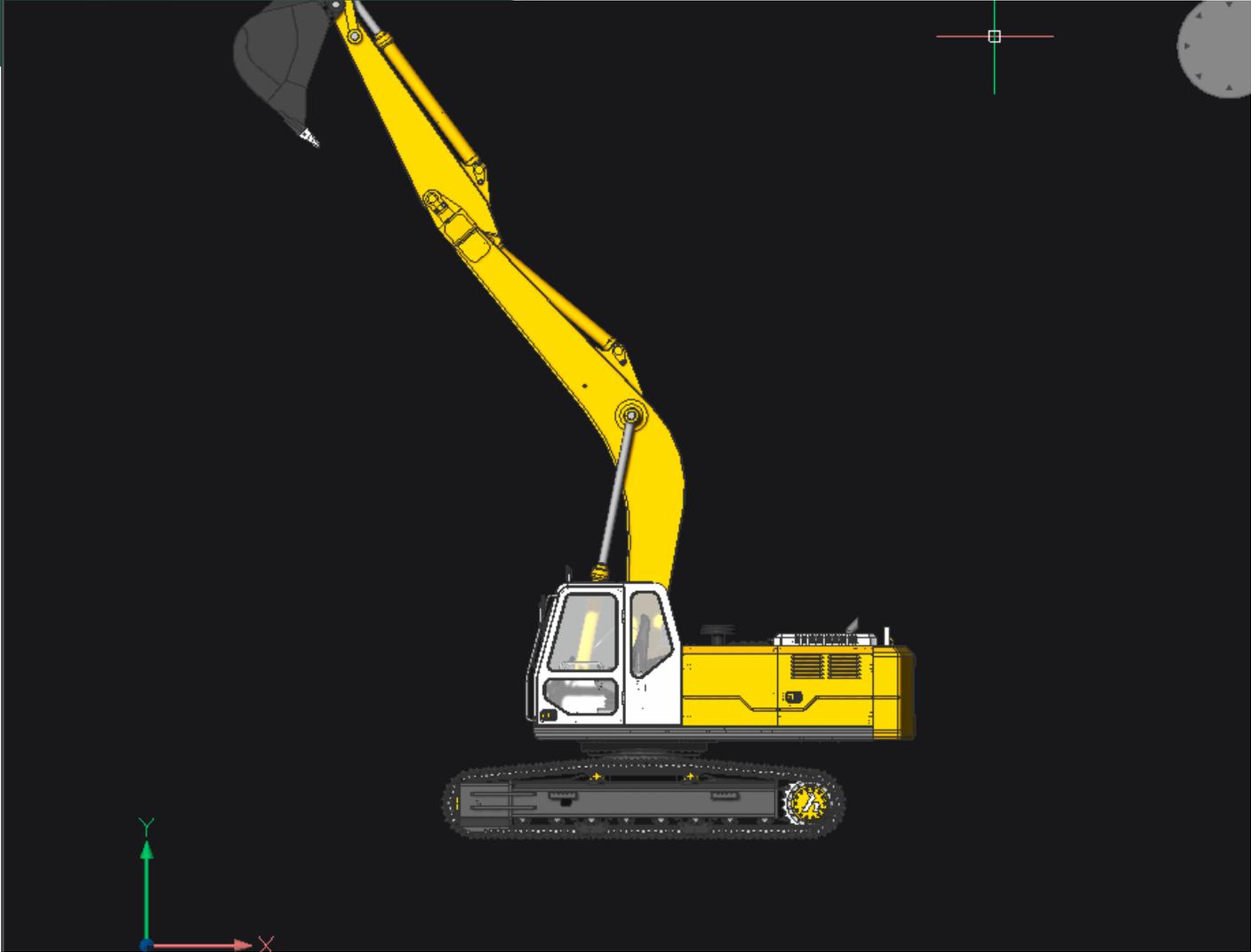
N* Новосибирский
государственный
университет
***НАСТОЯЩАЯ НАУКА**

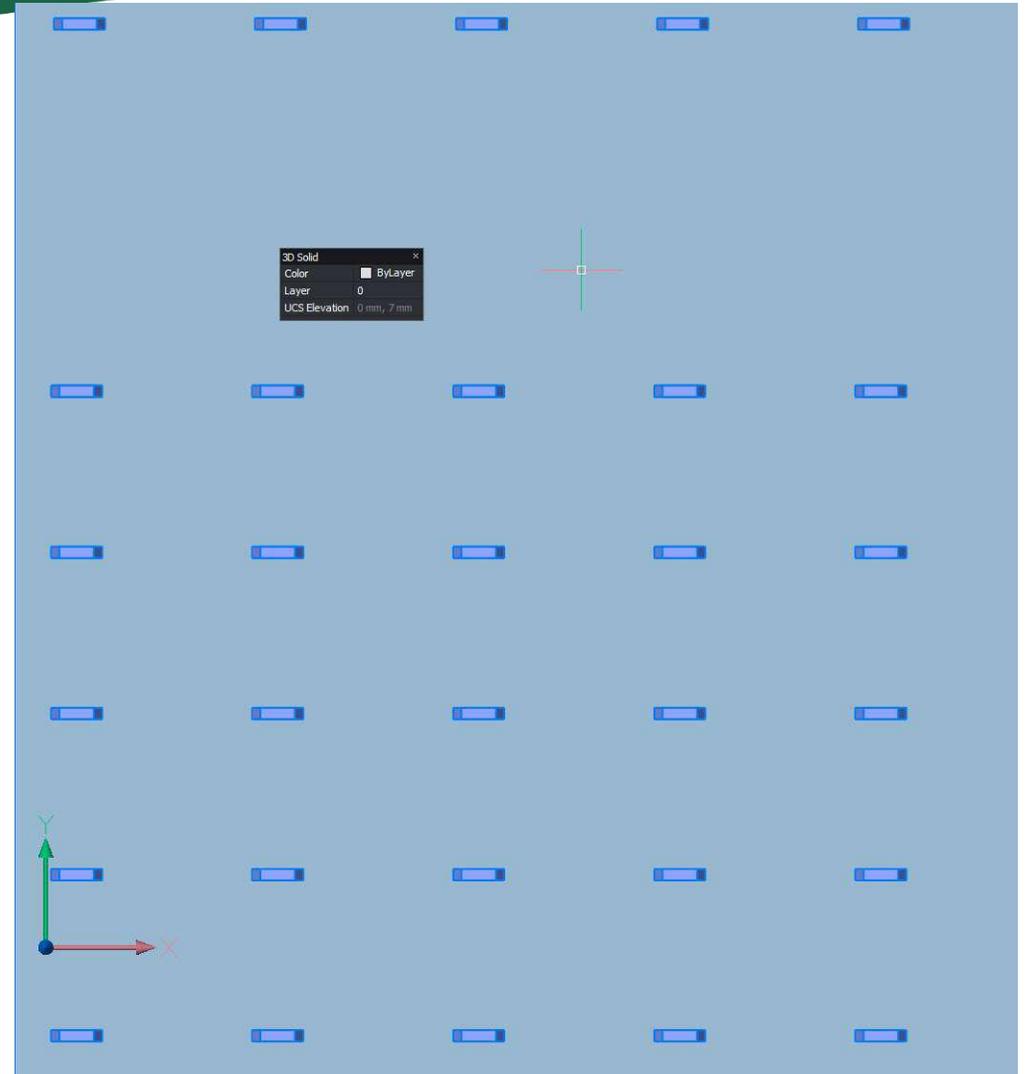
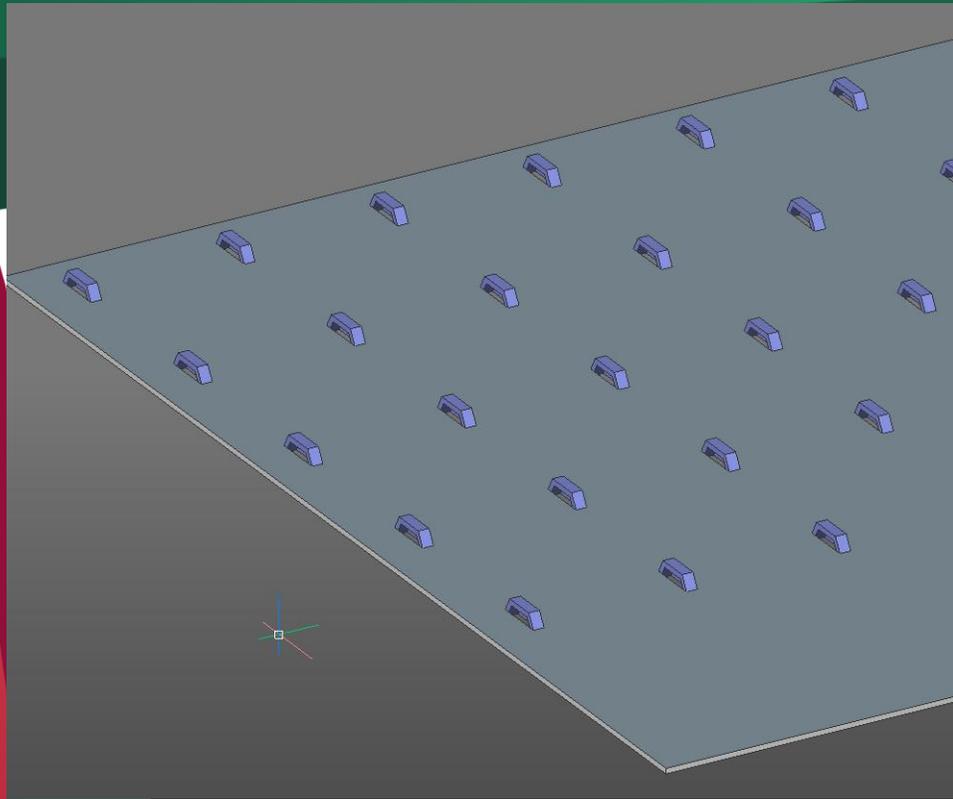
Крупская Вероника Александровна, ПМ-83, +7996-543-26-21

Малышкин Виктор Эммануилович, д.т.н., профессор

*ОПТИМИЗАЦИЯ АЛГОРИТМОВ МОДУЛЯ
LGSEP CAD-СИСТЕМЫ BRICSCAD*

Новосибирск 2022





В данном случае для тестирования производительности был использован процессор на микроархитектуре CoffeeLake 9 поколения, i7-9750H, который поддерживает векторное расширение AVX2.

Для вычисления поворота ковша используется итерационный алгоритм Ньютона-Рафсона. Метод итерационный и оптимизации с помощью параллельных вычислительных технологий над ним не будут тривиальными. Поэтому было решено сосредоточиться на более очевидных оптимизациях.

Время вычислений в миллисекундах от количества потоков.

