

Векторизация алгоритмов анализа динамического напряженно- деформированного состояния для архитектур x86 с поддержкой векторных регистров

В.В. Гетманский, А.Е. Андреев, Е.С. Харьков, Е.О. Мовчан

Волгоградский государственный технический университет

кафедра ЭВМ и С

Вычислительный кластер ВолгГТУ – неоднородный вычислительный комплекс на базе x86 Xeon и MIC

Вычислительный кластер
ВолгГТУ на базе Intel Xeon E5 /
Intel Xeon Phi / KNL – Топ-50
(40, март, 47 – сентябрь)

(24 nodes – >39 TFlops on Intel
HPL with offload, E5, KNC and
KNL used)

На Linpack. А что на
реальных задачах ?..



Вычислительный кластер ВолгГТУ – неоднородный вычислительный комплекс на базе x86 CPU и MIC

Четыре и более типов вычислителей (+ FPGA)

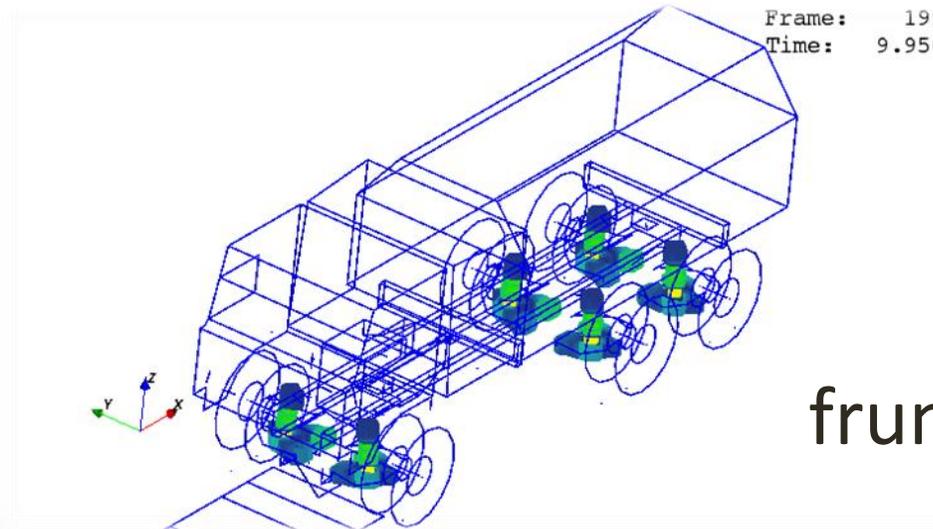


OpenMP + MPI + SIMD

Программный комплекс ФРУНД

(формирование и решение уравнений нелинейной динамики)

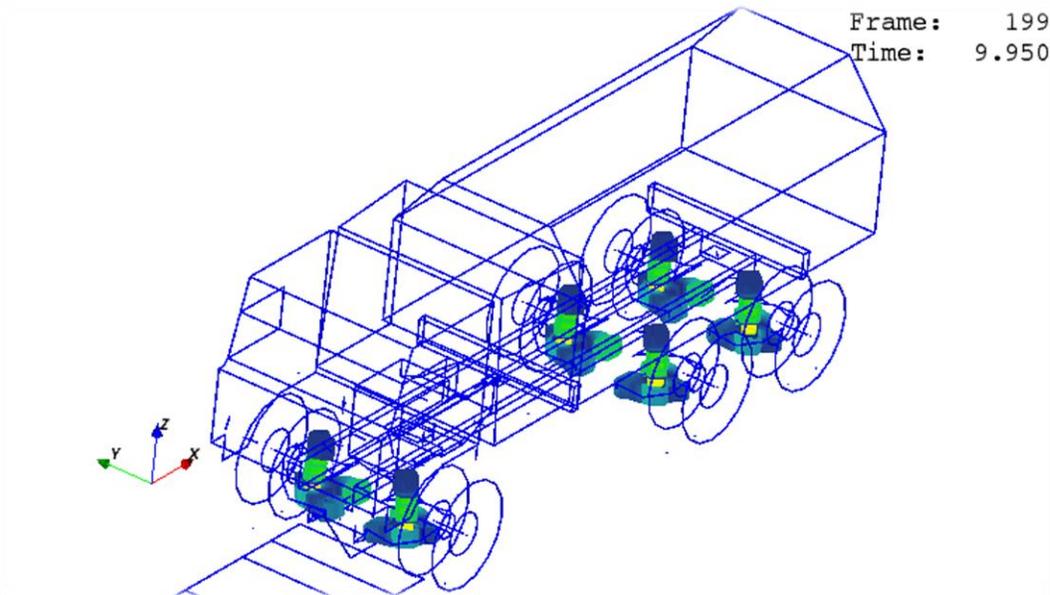
Пакет относится к САЕ системам, использующим т.н. многотельные модели (MBS) представления сложных технических систем, а также междисциплинарные расчеты для анализа их динамических характеристик. Применяется для решения задач из разных областей, включая создание новых транспортных систем, спецмашин, управление шагающими машинами, включая антропоморфных роботов.



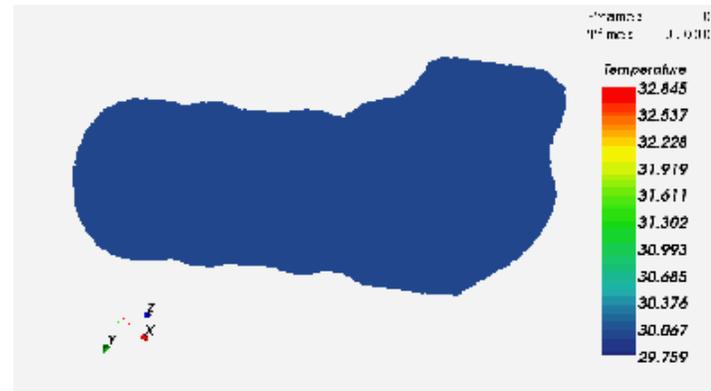
frund.vstu.ru



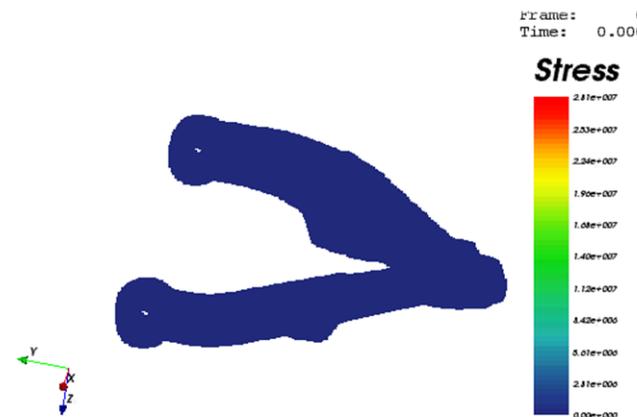
Междисциплинарное моделирование в пакете ФРУНД



Движение по неровной поверхности



Процесс теплопередачи (нагрев тела)



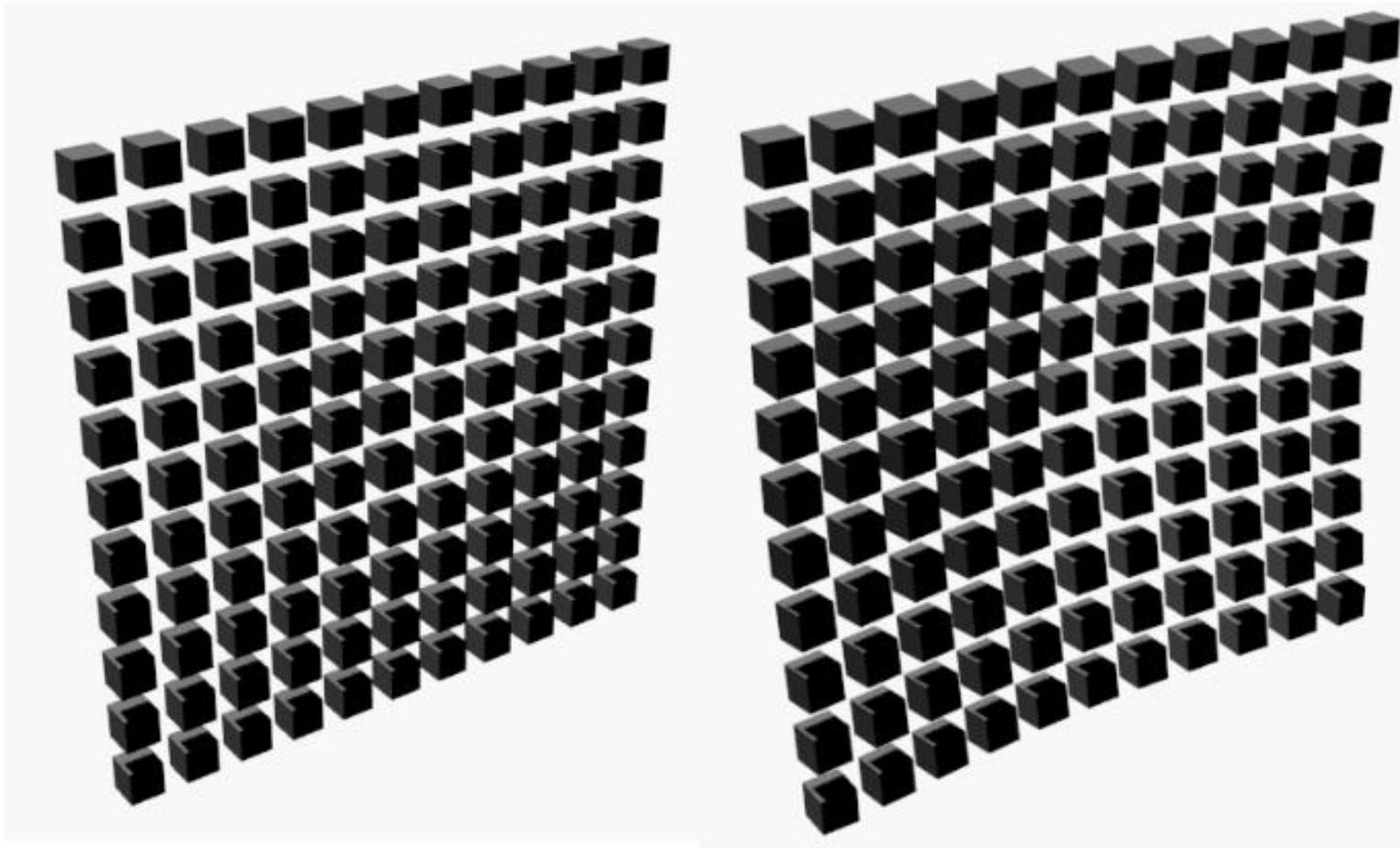
Напряжения при деформации тела (расчет НДС)

Общая постановка задачи – динамика систем тел

- Задача динамики (постановка в форме системы уравнений Лагранжа 1-го рода, программный комплекс ФРУНД)

$$\begin{cases} \mathbf{M}\ddot{\mathbf{x}} + \mathbf{D}_x^T \boldsymbol{\alpha}(2\boldsymbol{\mu}\boldsymbol{\Omega}\dot{\mathbf{D}} + \boldsymbol{\Omega}^2 \mathbf{D}) + \mathbf{D}_x^T \boldsymbol{\lambda} = \mathbf{q}(\dot{\mathbf{x}}, \mathbf{x}, t), \\ \mathbf{D}_x \ddot{\mathbf{x}} = \mathbf{h}(\dot{\mathbf{x}}, \mathbf{x}, t). \end{cases}$$

Дискретное представление тел для расчета НДС



Дискретные элементы описываются ДУ вида :
$$M\ddot{y} = q(\dot{y}, y, t) - Ma(t) + s(\dot{y}, y)$$

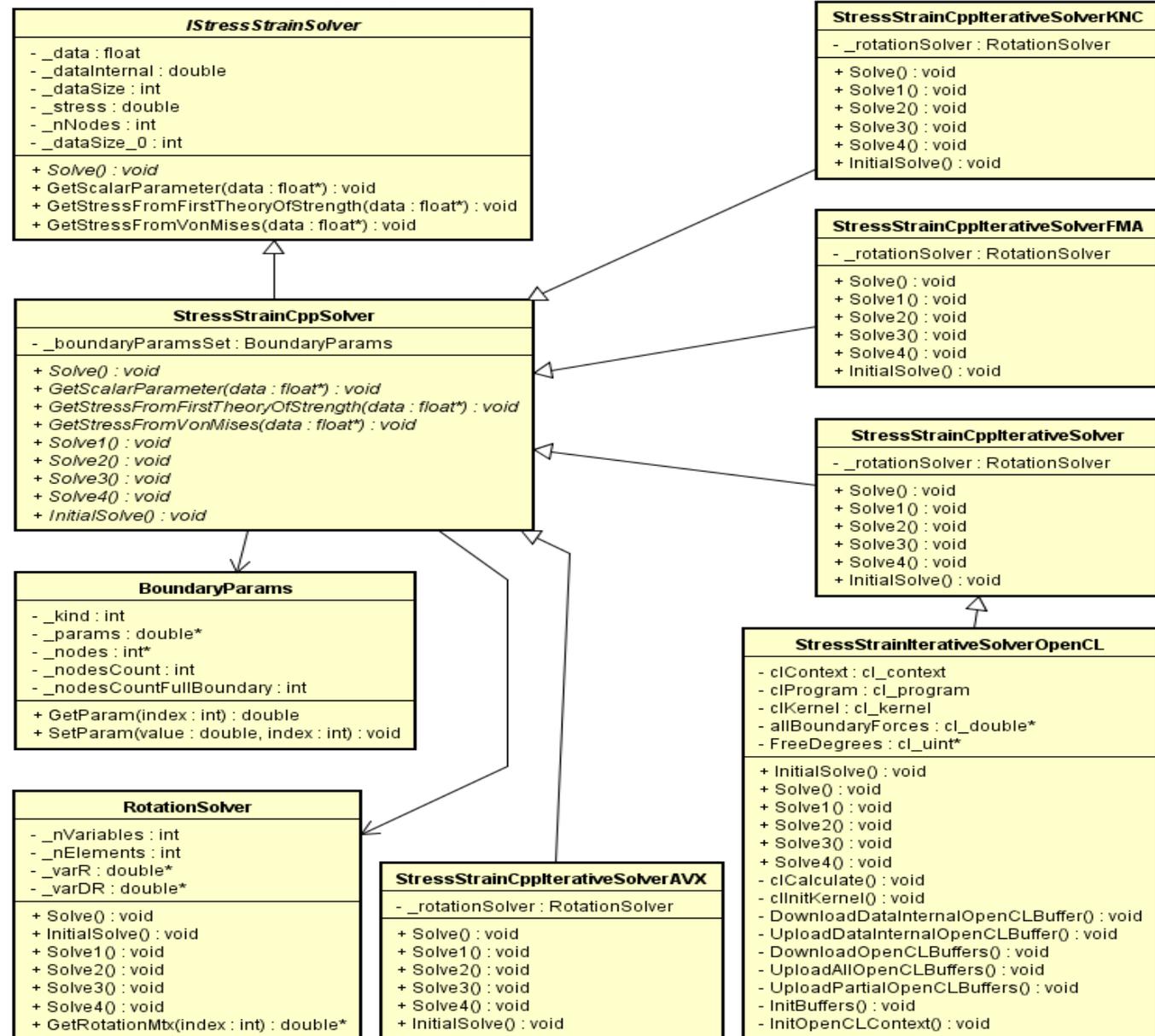
Модернизация и развитие пакета ФРУНД

- Внедрение параллелизма на разном уровне
- Разработка нового пред-пост процессора
- Дополнительные модули (например, для управления актуаторами роботов)
- Переработка кода (исходные решатели написаны на Фортране достаточно давно – унаследованный код) – **переход к ОО – версии (тепловой решатель, решатель НДС)**

00-версия решателя НДС для FRUND

Новая версия FRUND,
разрабатываемая на C++.

- + Модульность
- + Масштабируемость
- + Портруемость
- + Гибкость применения



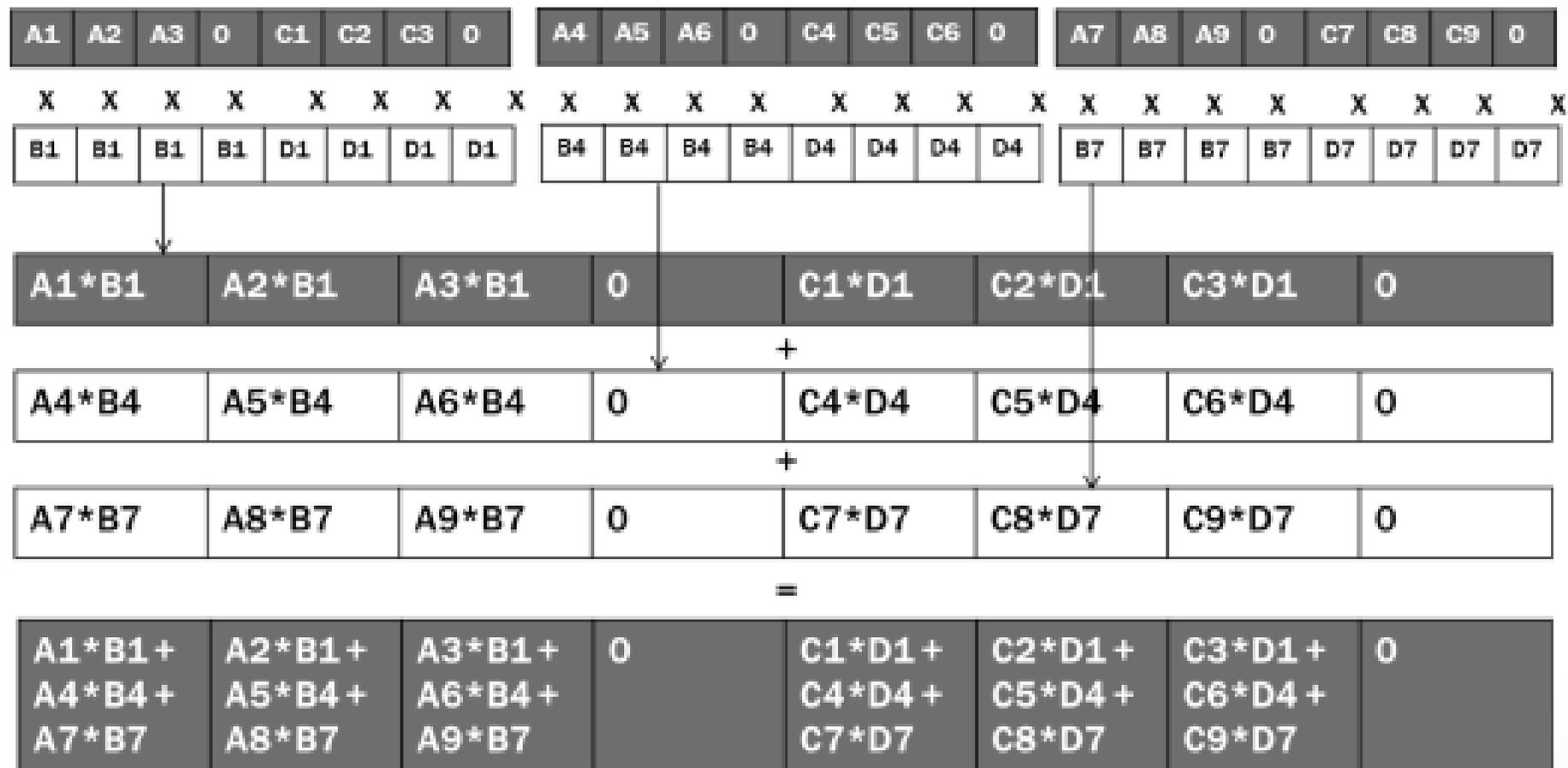
Параллелизм в пакете ФРУНД

- Декомпозиция и параллельный расчет сложной многотельной модели (MPI, OpenMP) – ограничение масштабируемости, сверхбольшие модели встречаются редко
- Параллельный междисциплинарный расчет (динамика, деформация, нагрев)
- Расчет тепловых задач (нагрев узлов), напряженно-деформируемого состояния на CUDA / OpenCL
- **Векторизация решателей ФРУНД**
 - **Интегрирование СОДУ методом РК4 (простая арифметика над векторами)**
 - **Вычисление правых частей (матричная алгебра)**

Опыт «ручной» векторизации

- Для вычислительного ядра пакета FlowVision – ускорение базовых операций до 11 раз на float (AVX + FMA)
- Для системы астрофизического моделирования AstroPhi / CosmoPhi (до 6-7 раз для архитектуры MIC)
- *Опыты с переносом алгоритма сравнения шинглов на MIC и векторизация – ускорение до 15 раз*

Одновременная обработка 2 пар матриц на 512-битных регистрах (пример)



Вариант реализации с помощью blend

```
zmm08=_mm512_extload_ps(vec+ 0, _MM_UPCONV_PS_NONE,  
    _MM_BROADCAST_1X16, _MM_HINT_NONE);  
zmm09=_mm512_extload_ps(vec+ 4, _MM_UPCONV_PS_NONE,  
    _MM_BROADCAST_1X16, _MM_HINT_NONE);  
zmm10=_mm512_extload_ps(vec+ 8, _MM_UPCONV_PS_NONE,  
    _MM_BROADCAST_1X16, _MM_HINT_NONE);  
zmm11=_mm512_extload_ps(vec+ 12, _MM_UPCONV_PS_NONE,  
    _MM_BROADCAST_1X16, _MM_HINT_NONE);  
  
zmm06 = _mm512_mask_blend_ps(mask32_0, zmm08, zmm09);  
zmm07 = _mm512_mask_blend_ps(mask32_1, zmm10, zmm11);  
zmm04 = _mm512_mask_blend_ps(mask32_2, zmm06, zmm07);
```

Вариант реализации с помощью swizzle

```
zmm04 = _mm512_swizzle_ps(zmm08, _MM_SWIZ_REG_AAAA);  
zmm05 = _mm512_swizzle_ps(zmm08, _MM_SWIZ_REG_BBBB);  
zmm06 = _mm512_swizzle_ps(zmm08, _MM_SWIZ_REG_CCCC);  
zmm07 = _mm512_swizzle_ps(zmm08, _MM_SWIZ_REG_DDDD);
```

Результаты – Xeon E5 с FMA (1 ядро)

1 ядро	XEON auto, t,c	XEON FMA, t,c	Ускорение
Повороты	1,69	1,69	1,00
правые части	33,4	15,33	2,18
Интегрирование	0,65	0,44	1,48
Всего	35,74	17,46	2,05

Результаты – Xeон Phi KNC (1 ядро)

1 ядро	KNC auto, t, c	KNC AVX512, t, c	KNC AVX512-2, t, c	Ускорение
повороты	337,22	339,11	338,25	1,00
правые части	438,1	196,44	139,32	3,14
интегрирова ние	10,6	3,42	3,51	3,02
Всего	785,92	538,97	481,08	1,63

Результаты – Xeon Phi KNL (1 ядро)

1 ядро	KNL auto, t, c	KNL FMA, t, c	KNL AVX- 512, t, c	Ускорение
Повороты	7,4	7,4	7,4	1,00
правые части	131,9	61,7	52,9	2,49
Интегриров ание	3,77	0,76	0,61	6,18
Всего	143,07	69,86	60,91	2,35

Результаты – Xeon E5 FMA (10 ядер)

все ядра/потоки	XEON auto 10/1, t, c	XEON FMA 10/10, t, c	XEON, ускорение
повороты	1,69	1,11	1,52
правые части	33,4	2	16,70
интегрирование	0,65	0,06	10,83
всего	35,74	3,17	11,27

Результаты – Xeon Phi KNC (60/240 потоков)

все ядра/потоки	KNC auto 1, t, c	KNC AVX512 60/240, ускорение	KNC AVX512 60/60, ускорение
повороты	337,22	21,8	18,01
правые части	438,1	6,52	7,27
интегрирование	10,6	0,09	0,13
всего	785,92	28,41	25,41

Результаты – Xeon Phi KNL (64 потока)

все ядра/потоки	KNL auto, t, с	KNL FMA 64/64, t, с	KNL AVX512 64/64, t, с	KNL, ускорение
повороты	7,4	3,3	3,3	2,24
правые части	131,9	1,47	1,43	92,24
интегрирование	3,77	0,06	0,08	47,13
Всего	143,07	4,83	4,81	29,74

Xeon vs Phi KNC / Phi KNL

	KNC / Xeon E5	KNL / Xeon E5
повороты	16,23	2,97
правые части	3,64	0,72
интегрирование	2,17	1,33
всего	8,02	1,52

Перспективы развития

- переход на float (требует исследования – расчет частично нуждается в double) – позволит использовать больше элементов в 512-битных регистрах
- переработка алгоритма для более эффективного использования памяти (работа с соседними адресами с сохранением преимуществ векторной обработки)
- оптимизация / изменение алгоритма координатных преобразований
- совместный расчет на разных вычислительных узлах (CPU, CPU + KNC, 2xCPU +2xKNC, KNL) для больших моделей, для реализации в реальном решателе необходимо доработать систему декомпозиции и синхронизации моделей
- реализация версии решателя только для Xeon Phi (native режим, разгрузка CPU).

Спасибо за внимание !

Контакты:

зав. каф. ЭВМ и С ВолгГТУ
к.т.н. Андреев Андрей Евгеньевич
e-mail: andan2005@yandex.ru

к.т.н., инженер компании Intel
Гетманский Виктор Викторович
e-mail: victor.getmanski@gmail.com

инженер компании Intel, магистрант
каф. ЭВМ и С ВолгГТУ
Харьков Егор Сергеевич
e-mail:

магистрантка каф. ЭВМ и С ВолгГТУ
Мовчан Евгения Олеговна
e-mail:

Работа выполнена при поддержке РФФИ
и администрации Волгоградской области
грант № № 16-47-340385.