

**Санкт-Петербургский
государственный
политехнический
университет**

**IV Международная
молодежная школа
«Высокопроизводи
тельные
вычисления на
GRID-системах»
Северный
(Арктический)
федеральный
университет имени
М.В. Ломоносова**

05 февраля



Болдырев Юрий Яковлевич

**«Суперкомпьютерные технологии в
промышленности, науке и образовании»**

Начну с вопроса!

Вопрос. Что или какие технологии лежат в основе применения компьютеров (суперкомпьютеров) как инструментария?

Цель лекции:

– ознакомление с фундаментальными основами современных суперкомпьютерных технологий и их ролью и местом в системе технологий, которыми владеет человечество, имея в виду промышленность, науку и образование.

Такая цель достигается путем рассмотрения **следующих основных задач:**

- изучение основ современной концепции математического моделирования и ролью и местом в ней компьютерных (суперкомпьютерных) систем и технологий;

- рассмотрение понятия (определения) суперкомпьютера и суперкомпьютерных технологий ;

- ознакомление с историей развития программирования – от программирования в кодах до программных комплексов для инженерного и естественнонаучного анализа;

- роль и место компьютерных (суперкомпьютерных) технологий в предметной системе технологий (промышленность, науку и образование).

Часть 1 Современная концепция математического моделирования её существо и развитие, и роль и место в ней компьютерных (суперкомпьютерных) систем и технологий.

Сначала изложим современную Концепцию Математического моделирования в её существе.

Начнем с того, что математическое моделирование в течение XX века превратилось в тотальный инструментарий, которым владеет человечество.

Вообще, современное понимание существа математического моделирования начало формироваться в конце XIX начале XX веков. В его становлении выдающуюся роль сыграли труды Р. Фреше (1878-1973) и Д. Гильберта (1862-1943), которые ввели новое понимание близости в математике (соответственно, метрическое и гильбертово пространства). В итоге были созданы новые подходы в вычислительной математике и заложены теоретические основы современного математического моделирования. В дальнейшем в его формировании решающую роль сыграли новые идеи в формулировке задач математической физики в форме интегральных тождеств, а также восходящий к Р.Куранту (1888-1972) метод конечных элементов, послуживший основой разработки вариационных и проекционных разностных методов решения задач математической физики.

Огромную роль в формировании современной концепции математического моделирования сыграла Российская наука в лице **А.А. Самарского (1919-2008), О.М. Белоцерковского (р. 1927)** и других выдающихся ученых.



Отметим, что математическое моделирование, оказало по своему существу не меньшее, а скорее даже большее влияние на достижения цивилизации, чем революция в физике, восходящая к теории А.Эйнштейна и работам великих физиков: Э. Резерфорда, Н.Бора и других, работавших на рубеже XIX - XX веков.

Действительно, замена эксперимента физического, который стоял во главе угла во все века в исследовательской и инженерной деятельности, на эксперимент вычислительный (которым по своей сути и является математическое моделирование) привел к революционной смене методологии в изучение природы и развитии техники.

Итак, приведем поблочное описание Концепции Математического моделирования, следуя А. А. Самарскому.

I блок. *Составление математической модели явления, процесса, задачи и т.д.* Это, вообще говоря, предметная область естественных и других наук, включая и социальные, где составляются **количественные соотношения** для описываемых явлений и процессов. И это труднейшая область научной деятельности.

Мы знаем законы Ньютона, выражаемые количественно, уравнения Максвелла электродинамики и так далее. Отметим лишь, что в М.М. рассматриваемый блок один из самых критикуемых и обсуждаемых, поскольку с ним связана не только простота или сложность описания изучаемого явления, но и каждый раз свое понимание исследователем глубины этого описания.

Итак, выделяя самое основное, блок - это некоторая «замена» реального физического или иного явления его математической моделью, то есть описание всех его свойств-характеристик с помощью математических соотношений. Эта идея оказывается чрезвычайно плодотворной, особенно при изучении сложных явлений и конструировании высокотехнологичных систем машин и приборов.

Например, исследование аэродинамики самолета в аэродинамической трубе, при больших скоростях скорости 200 м/с практически невозможно, но мы можем создать математическую модель его обтекания, получив характеристики этого обтекания с помощью компьютерного эксперимента. В этом, вычислительный эксперимент позволяет исследовать обтекание самолета, которые относятся к гиперзвуковым (более 2-3 км/с), какие принципиально невозможно получить в натурном эксперименте в аэродинамической трубе.



Однако, проблема полномасштабного применения методов математического моделирования является весьма многоплановой и имеет много трудных сторон.

Стоит задуматься, например, над вопросом точного описания явления с помощью количественных (математических) соотношений. Как точно мы описываем то или иное явление? Для одного круга проблем достаточно выявления главных факторов описывающих процесс, для другого требуется гораздо более тонкое и глубокое описание явления.

II блок. *Анализ математической корректности построенной математической модели, описывающей нашу задачу.*

Весь спектр рассматриваемых здесь проблем носит сугубо математический характер и в подавляющем большинстве обходится исследователями. У опытного исследователя всегда есть понимание, «чутье» если угодно на то, что должно получиться. Однако такой путь весьма опасен, особенно для начинающих, и примеров, которые показывают к чему это может привести, имеется довольно много.

Путь отрицания анализа корректности в постановке задачи особенно опасен, при слепой вере в то «что компьютер все посчитает». Всегда желательно обратиться к профессионалам математикам, которые скажут Вам, есть ли в задаче какие - то проблемы.

Приведем пример. Пусть имеется система из двух линейных алгебраических уравнений:

$$1.0x_1 + 0.9x_2 = 1.9$$

$$0.9x_1 + 0.8x_2 = 1.7$$

Очевидно, что решением этой системы являются $x_1 = x_2 = 1$. Однако, $x_1 = 3$ и $x_2 = -1,22$ являются решением следующей системы:

$$1.0x_1 + 0.9x_2 = 1.902$$

$$0.9x_1 + 0.8x_2 = 1.724$$

то есть малое изменение

$$\delta b = (0,002; 0,024)^T$$

в исходных данных задачи, а именно в столбце правой части

$$b = (1,9; 1,7)^T$$

привело к очень большой ошибке в результате

$$\delta x = (2,0; -1,22)^T$$

Попытаемся выяснить, в чем дело, почему малые изменения исходных данных привели к столь большим изменениям решения?

В данном случае ситуации можно придать наглядный геометрический смысл (рис. 1). Действительно, как мы знаем из курса алгебры, решение системы из двух линейных уравнений находится как точка пересечения соответствующих линий:

$$x_2 = (-1.0x_1 + 1.9) / 0.9,$$

$$x_2 = (-0.9x_1 + 1.7) / 0.8$$

Как видим наши линии, пересекающиеся в точке $(1, 1)$ почти параллельны. Поэтому малые изменения коэффициентов определяющих эти линии приводит к сильному изменению положения точки их пересечения.

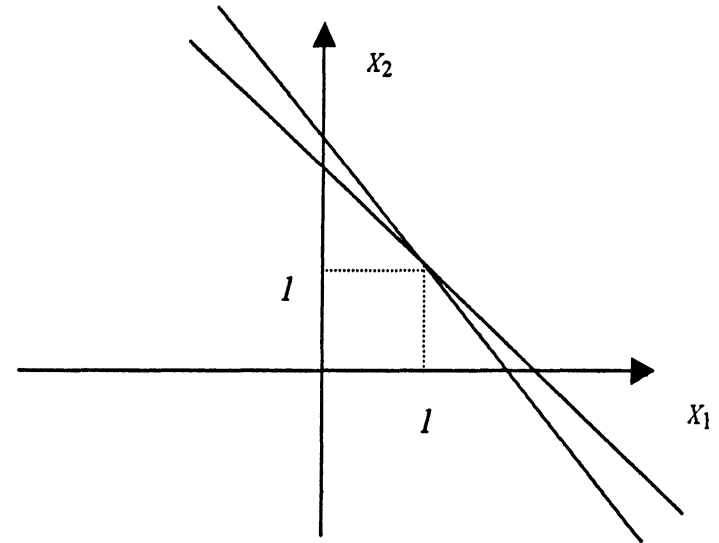


Рис. 1

Вообще решение проблемы корректности той или иной задачи включает в себя решение следующих ключевых задач:

- существования решения (решений) в том или ином классе (векторов, функций, целых или вещ. чисел и т.д.);
- единственность решения;
- устойчивость решения по отношению к возмущению параметров задачи (в нашем примере это возмущение правой части).

III блок. *Переход от непрерывной математической модели к модели дискретной.*

Отметим, предварительно, что в ряде задач этот блок может отсутствовать, то есть, он, как правило, характерен для таких моделей - явлений, которые описываются дифференциальными и интегральными соотношениями.

При переходе от непрерывной модели к дискретной мы, по сути дела, переформулируем нашу задачу и получаем некоторую новую в которой присутствует один или несколько параметров, характеризующих новую дискретную задачу и которых не было в исходной задаче.

Простым примером этого являются задача вычисления интеграла, когда мы вычисляем его с помощью формул трапеций или Симпсона, в которых появляется параметр N , как раз и характеризующий степень точности такого приближения.

Приведем пример, наглядно демонстрирующий эту новую формулировку задачи. Пусть требуется решить задачу Коши):

$$y' = -y, \quad x \in [0, 1], \quad \text{при условии: } y(0) = 1. \quad (\text{A})$$

Упростим задачу - пусть требуется найти только значение $y(1)$.

Построим вычислительную процедуру, опираясь на определение производной

$$y'(x_i) = \lim(y(x_i + \Delta x) - y(x_i)) / \Delta x, \quad \text{при } \Delta x \rightarrow 0,$$

и зафиксируем величину Δx рассматривая соответствующее приближение для производной.

Этому Δx отвечает такое разбиение промежутка $[0, 1]$: $\Delta x = 1/N$, например, на N одинаковых шагов (частей). Тогда (А) приближенно можно записать в виде

$$y(x_i + \Delta x) - y(x_i) = -y(x_i)\Delta x \quad (i=0, 1, \dots, N-1)$$

или

$$y_{i+1} = y_i(1 - \Delta x), \quad i=0, 1, \dots, N-1 \quad (\text{B})$$

Согласно: $y(0) = 1$ и система уравнений (Б), разрешается т. о.:

$$y_1 = y_0(1 - \Delta x) = 1 - \Delta x, \quad y_2 = y_1(1 - \Delta x) = (1 - \Delta x)^2, \dots, \quad y_{i+1} = y_i(1 - \Delta x) = y_{i-1}(1 - \Delta x)^2 = \dots = (1 - \Delta x)^{i+1}$$

Откуда для искомого $y_N = (1 - \Delta x)^N = (1 - 1/N)^N$ имеем $y \approx e^{-1}$, что является решением исходной задачи Коши при $x=1$. Понятно, что это решение мы могли написать и сразу, здесь же нам важно не это, а то, что мы его нашли «явно» как предельное для численного решения.

Чем оно характеризуется? Оно получено из новой задачи (Б), в которой присутствует некоторый параметр N , - которого не было в исходной задаче (А).

И, как мы видим, при $N \rightarrow \infty$ решение новой (приближенной) задачи стремиться к решению нашей исходной.

IV блок. Анализ математической корректности вновь полученной дискретной задачи (на основе исходной)

По сути дела это и есть предмет вычислительной математики. Здесь рассматриваются те же проблемы, что и в блоке II, к которым добавляется еще одна задача, - весьма важная.

Эта задача должна ответить на такой вопрос: - будет ли при параметре $N \rightarrow \infty$ (или для их совокупности) дискретная задача, построенная в блоке III, стремиться к исходной непрерывной задаче?

V блок. *Написание алгоритма для дискретной задачи, то есть последовательности вычислительных шагов и его перенос на вычислительную систему, - программирование (или кодирование).*

Этот блок носит совершенно особый характер и его роль, со времени внедрения вычислительных машин в научную и инженерную практику, непрерывно менялась и как видится и будет меняться.

Вообще эта область Знания, сегодня относится к *программному обеспечению* вычислительных машин или используя принятую терминологию к области «*software*». Сегодня это огромный пласт прикладной математики, где на первый план выдвигаются так называемые наукоемкие программные системы (подробнее об этом далее).

VI блок. *Отладка программы (или настройка программной системы на нашу задачу), то есть её тестирование, получение результатов и их анализ.*

Изложенная схема, конечно, является очень общей и её применение в полном объёме отдельным исследователем при решении какой либо задачи, в общем, не характерно и является, как правило, функцией научной группы или коллектива.

Безусловно, прорывом в широчайшем внедрении Математического моделирования в инженерную и научную практику стало появление в начале второй половины XX века ЭВМ и сопутствующих технологий. Сегодня они играют главенствующую роль в проникновении Математического моделирования во все отрасли жизни – от науки до искусства и т.д. При этом использование суперЭВМ позволяет решать такие классы задач, к которым не могли даже подступиться исследователи и инженеры предыдущих поколений.

Итак два ключевых для дальнейшего тезиса

Тезис 1-й. Математическое моделирование на основе компьютерных (суперкомпьютерных) технологий превратилось в **тотальный** инструментарий высокоразвитых экономик, охватив науку, промышленность, социальную и иные сферы.

Тезис 2-й. Сегодня человечество, принципиально, перешло ту грань за которой оно способно решать **ЛЮБЫЕ** корректно поставленные задачи. (Если кому-то это высказывание покажется чрезмерным, можно указать на то, что сегодня решены или решаются такие классы задач, сама физико-математическая постановка которых была немыслима 15-20 лет тому назад. А что будет через 20 или 50 лет!?)

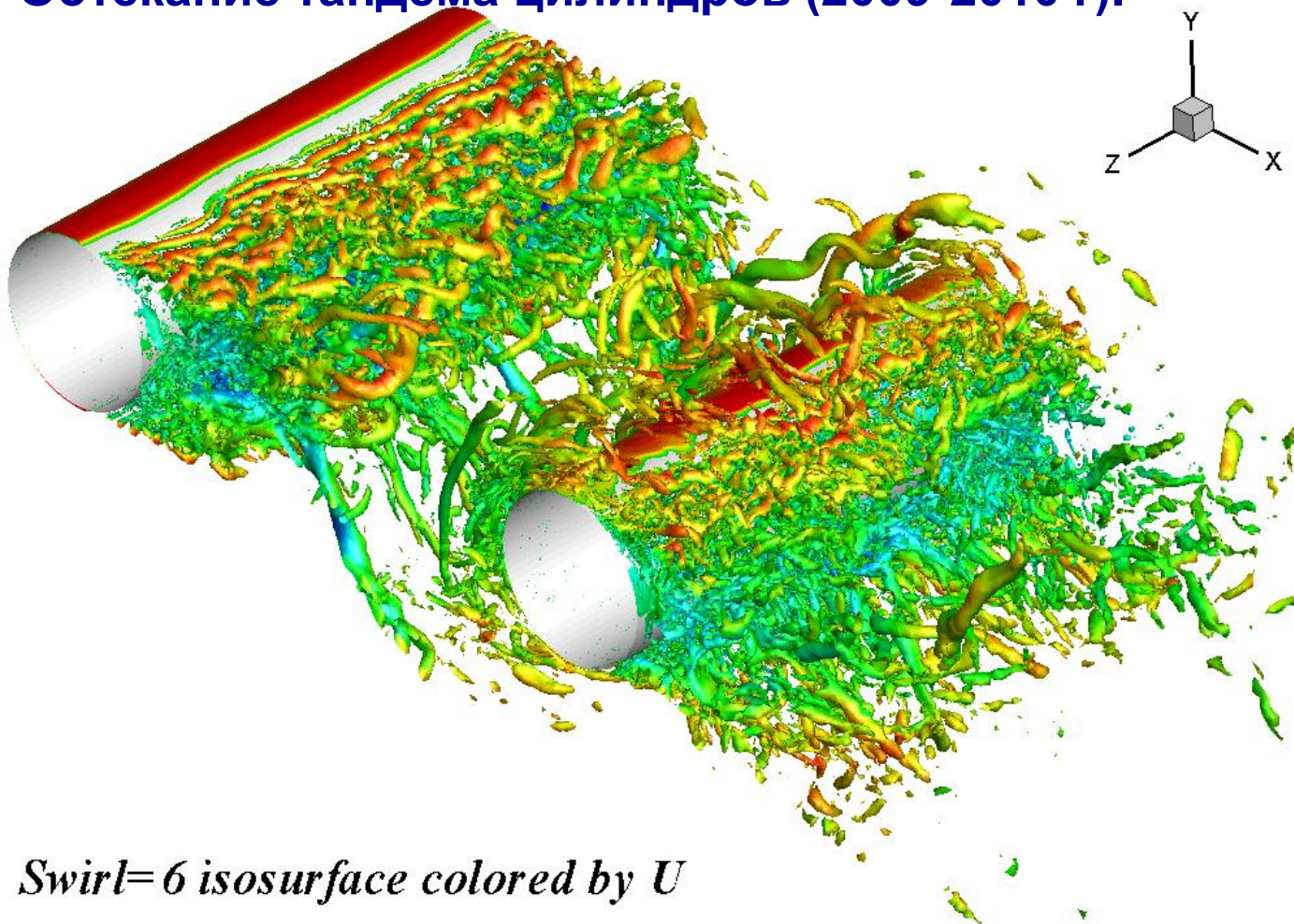
Приведем характерный пример достижений в исследовании Природы. Этот пример подтверждает что, суперкомпьютерные технологии являются уникальным инструментом её изучения.

Математикам, физикам и механикам хорошо известна одна из нерешенных проблем XX века – проблема турбулентности. По своей физико-математической природе это проблема корректного математического моделирования движения вязкой жидкости или газа при сравнительно больших (примерно, больших 1000) числах Рейнольдса Re ($Re = V \times L / \nu$, где L и V характерные размеры области течения и его скорость, а ν – кинематическая вязкость жидкости).

Определенная теоретическая нерешенность проблемы турбулентности привела к потребности и необходимости в развитии методов математического моделирования для её исследования. Здесь, уместно отметить, что исторически численное моделирование турбулентности - одна из наиболее характерных задач применения суперкомпьютеров.

Задача представленная ниже решена в Лаборатории «**Вычислительная гидродинамика и турбулентность**» СПбГПУ, руководитель проф. **М.Х.Стрелец**.

Обтекание тандема цилиндров (2009-2010 г).



Поле скорости поперечного обтекания тандема цилиндров при числе $Re = (V_0 D) / \nu = 1.4 \cdot 10^5$ (V_0 – скорость набегающего потока, D – диаметр цилиндра, ν – кинематическая вязкость).

Визуализация наглядно демонстрирует наличие как крупных вихревых структур, так и мелко-масштабной турбулентности.

Swirl=6 isosurface colored by U

Рис.2

Задача считалась на суперкомпьютере «Intrepid» Blue Gene/P Лаборатории Argonne National Lab (США) на 8160 узлах. Каждый узел содержит два 4-х ядерных процессора. В расчетах было использовано от 16320 до 65280 процессорных ядер! Сетка, содержит примерно 60 миллионов узлов. Характерный расчет шел около 11 суток.

Пример показывает эффективность математического моделирования на основе суперкомпьютерных технологий применительно к моделированию турбулентности. Сейчас такие подходы основаны на:

- **численном решении осредненных уравнений Навье-Стокса**, т.н. уравнениях Рейнольдса (**RANS** *Reynolds Average Navies-Stokes*), замкнутых с помощью тех или иных полуэмпирических моделей турбулентности;
- **прямом численном моделировании турбулентности** (**DNS** *Direct Navies-Stokes*) - непосредственное интегрирование нестационарных трехмерных уравнений Навье - Стокса с разрешением всего спектра пространственно - временных масштабов турбулентности;
- **методах, основанных на «точном» разрешении лишь части спектра** и моделировании остальных масштабов (метод моделирования крупных вихрей (**LES** *Large Eddy Simulation*) и гибридные RANS/LES методы в первую очередь Метод моделирования отсоединенных вихрей (**DES** *Detached-Eddy Simulation*), - этим методом решена приведенная задача.

Приведем оценку «готовности» **LES** и **DNS** методов для расчета обтекания самолета или автомобиля на основе весьма оптимистичного прогноза роста быстродействия компьютеров (в 2 раза каждые 5 лет).

| | Размер сетки | Число временных шагов | Готовность |
|------------|-------------------------------|------------------------------|-------------|
| LES | $10^{11.5}$ | $10^{6.7}$ | 2045 |
| DNS | 10^{16} | $10^{7.3}$ | 2080 |

А теперь давайте определимся с понятиями!

Вопрос. А как корректно определить такое понятие как суперкомпьютер?

И еще вопрос. Мы пользуемся понятием суперкомпьютерных технологий, а что это такое? Как их определить!

ИТАК, - к определению понятия суперЭВМ (или что мы под этим понимаем)

Вообще говоря точного определения понятия **суперЭВМ** нет!

Имеется несколько определений, носящих не вполне строгий характер.

Например.

Было такое «определение» суперкомпьютер – это вычислительная система стоимостью 1-2 млн. долларов (!?).

Или, - суперкомпьютер это вычислительная система, производительность которой всего на порядок менее необходимой для решения «современных задач» (!?). (Но что такое современная задача?).

Нелепее всего выглядит определение Оксфордского словаря вычислительной техники 1986 г. Оно предписывало суперкомпьютеру требование иметь производительность в 10 млн. арифметических операций в секунду! И не меньше!

Нам видится наиболее приемлемым определение суперкомпьютера как «сосредоточенной» вычислительной системы, которая **имеет производительность на 3-4 порядка более высокую, чем массово распространенные компьютеры.**

Теперь уместно указать на то, что мы понимаем под **суперкомпьютерными технологиями**. Это следующая «триада»:

1. Производство СуперЭВМ.
2. Разработка программного обеспечения для СуперЭВМ.
3. Совокупность знаний и технологий для решения задач пп. 1 и 2, а также для задач предметного использования СуперЭВМ.

В рамках лекции мы, попытаемся раскрыть содержание главным образом 2 и 3 пунктов этой триады. В первую очередь, в области программного обеспечения, где современные компьютерные (суперкомпьютерные) технологии продвинулись очень далеко, приняв форму наукоемких программных комплексов (систем). Для своего применения такие системы требуют глубоких физико-математических знаний, знаний вычислительной математики и конечно предметной, например, инженерной области, физики, механики, биологии и т.д.

Сегодня, решение очень большого числа задач оказывается возможным в рамках таких комплексов, в которых, говоря образно, «встроена» изложенная выше концепция математического моделирования. Но прежде чем об этом подробно говорить необходимо проследить откуда ВСЕ пошло!

Часть 2. Начало эпохи программирования-
программирование на языке ЭВМ (в кодах) и появление
языка **Fortran** – начало эпохи языков программирования
и алгоритмических языков.

Операционные системы *Windows* или другие, с которыми Вы знакомы непременный атрибут каждого персонального компьютера. Эти, очень мощные по возможностям и очень большие по объёму, занимающие сотни мегабайт, программы, можно назвать программами – оболочками.

Однако, в той луковице, которой для пытливого глаза представляет программное обеспечение Вашего персональный компьютер, они являются самым внешним «слоем». Под этим «слоем», как правило, имеется ещё несколько «слоёв-уровней» и самый первый из них это уровень команд микропроцессора. Именно из этих команд и состоит в конечном итоге каждая из программ выполняемых на компьютере. В одной программе может быть несколько тысяч или даже миллионов команд.

Сейчас же нам важно проследить обратную тенденцию, а именно как десятилетиями лучшие математики и программисты мира «наращивали» слою упомянутой «луковицы».

Или, говоря современным языком, посмотрим, как развивалось программирование от уровня команд-инструкций первых ЭВМ, до уровня современных мощных системных и прикладных программ для суперЭВМ. Для этого мы вновь должны вернуться на более чем 50 лет назад, обратившись к функциональной схеме первых ЭВМ приведенной на следующем слайде на рис. 3.

Функциональная схема первых ЭВМ

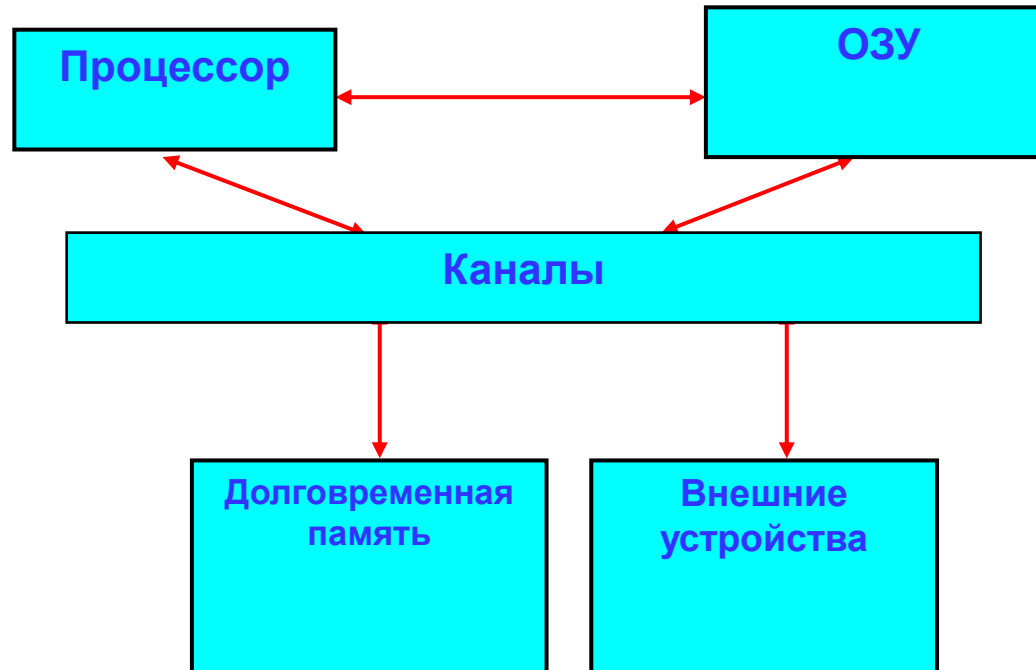


Рис. 3. Функциональная схема первых ЭВМ.

Все множество связующих магистралей ЭВМ обозначено **красными линиями**, причем стрелки указывают направление передачи информации.

Мы знаем, что в оперативной памяти хранятся как данные задачи (например, числа необходимые для вычислений), так и команды-инструкции, реализующие процесс вычислений, то есть **программа хода вычислений**. Отсюда и пошло наименование **программа!**

То есть процессор руководствуется в своих действиях, инструкциями – командами, «записанными» в ОЗУ. В этом и состоит **принцип программного управления ЭВМ**.

Но здесь не все так просто. Первые машины, например, БЭСМ-2М относилась к числу так называемых 3-х адресных ЭВМ. Это значило, что в процессе выполнения одной команды процессор (в общем случае) обращался к трем ячейкам оперативной памяти машины. Но были и одно и двухадресные машины.

Большинство команд, например, 3-адресной машины имели такой вид:

| | | | | |
|-------|-----|----|----|----|
| Адрес | КОП | A1 | A2 | A3 |
|-------|-----|----|----|----|

где

- Адрес – номер той ячейки ОЗУ где записана данная команда;
- КОП – номер выполняемой операции, например 001 – сложение, 002 – вычитание и т.д.;
- A1, A2 и A3 – адреса (номера) тех ячеек с которыми «работает» данная команда.

Например, команда

| | | | | |
|------|-----|------|------|------|
| 1023 | 001 | 0242 | 0351 | 0101 |
|------|-----|------|------|------|

записанная в ячейке ОЗУ 1023 означает сложение чисел из ячеек 0242 и 0351 и размещение итоговой суммы в ячейке с номером 0101.

Выше приведен пример 3-х адресной ЭВМ, какой была БЭСМ -2М, тогда как 2 – х и 1 – о адресные машины имели такой формат команд:

3-х адресная:

| | | | |
|-----|----|----|----|
| КОП | A1 | A2 | A3 |
|-----|----|----|----|

2-х адресная:

| | | |
|-----|----|----|
| КОП | A1 | A2 |
|-----|----|----|

1-о адресная:

| | |
|-----|----|
| КОП | A1 |
|-----|----|

Сегодня, как правило, компьютеры имеют команды переменной адресности, то есть в процессоре используются все типы команд приведенные на схеме.

А теперь задумаемся над вопросом.

Какого труда стоит написать программу на таком внутреннем языке команд компьютера?

Авторы первых программ выписывали на бланках тысячи команд в виде чисел, отвечающих как командам, так и адресам оперативной памяти. Представьте себе среднюю программу размером в 1000 трехадресных команд! Это же

$$1000 \times (4 + 3 + 12) = 19\,000 \text{ знаков!}$$

Даже самый опытный и аккуратный человек допустит ошибки, но ведь еще нужно переписать (закодировать) это на перфокарты!

Попробуйте переписать из любой таблицы подряд двадцать восемнадцатизначных чисел! И потом проверьте с друзьями ...

Понятно что такое программирование

- трудоемко, ненаглядно и порождает ошибки.

А ведь именно с этого и начиналось программирование!

Названные трудности тормозили внедрение ЭВМ в первые годы после их появления. Нужно было уйти от «массового» написания программ «в кодах», т.е. на языке команд ЭВМ. А как облегчить процесс программирования? Как переложить всю рутинную работу на «плечи» ЭВМ?

Схема, по которой здесь шла мысль исследователей, такова:

- любая математическая проблема (а только такие и рассматривались на первых этапах внедрения ЭВМ) может быть хорошо формализована;
- мы знаем все команды наших компьютеров;
- а почему бы не построить такую программу, которая на основе нашей формально записанной задачи строила бы программу на языке команд ЭВМ?

Но предварительно нужно записать на каком то близком к обычному математическому языку нашу задачу.

Такие языки, получившие наименование языков программирования начали создавать в самом начале 50 – годов. Здесь были и интересные отечественные разработки, а наибольшее распространение получил язык **FORTRAN** (Formula Translation). **Это был 1957 год!** Названные выше программы, - программы – переводчики стали называть трансляторами (в дальнейшем компиляторами, когда к ним добавились еще ряд функций).

Создание языка **FORTRAN**, породило «эпоху» проблемно ориентированных языков:

ALGOL 60, ALGOL 68 – алгоритмические языки, построенные на основе математической лингвистики были ориентированы в основном на очень широкий круг вычислительных задач;

PASCAL – язык программирования, ориентированный на широкий круг задач;

COBOL – язык который был создан для работы с большими массивами экономических данных;

LISP - язык созданный для работы с большими списками, и т.д.

Любопытно, что сам «прародитель» **FORTRAN** претерпел массу модификаций от **FORTRAN** к **FORTRAN 4** от **FORTRAN 77** к **FORTRAN 90** и, наконец к параллельной версии **FORTRAN**.

Сегодня многие пользуются языкам **C++**, впитавшим в себя все сильные стороны многих, языков, начиная с **FORTRAN**.

Часть 3. Начало эпохи массового внедрения компьютеров в научно-техническую сферу

Итак, в 1957 году был создан язык FORTRAN.

За ним как мы говорили появилась масса проблемно ориентированных языков – **ALGOL 60, ALGOL 68, ADA, LISP** и т.д.

Однако, непрерывно возрастающая сложность решаемых задач, приводила к значительным трудностям в «отладке» программ, реализующих их решение. На этом пути сформировалось понимание необходимости **выделения классов типовых задач.**

Так в инструментарий исследователей вошли пакеты стандартных прикладных программ. Сначала это были пакеты решения простых задач: - приближенное вычисление интегралов, решение алгебраических и трансцендентных уравнений, линейных систем и т.д. Затем уровень прикладных пакетов усложнился и они стали приобретать все более широкий характер в направлении решения задач по отраслям наук, например, в области механики жидкости и газа, механики твердого и деформируемого тела, биохимии и т.д. и т.п.

Быстро рос уровень сложности решаемых задач. Переход к пространственным задачам привел к необходимости решения проблем визуализации результатов вычислений. На этом пути был получен качественно новый подход к изучению механики, физики, биологии - стало возможным «просматривать» моделируемые процессы от появления микротрещин в металле, микро вихрей в жидкости и «конструирования молекул» с заданными свойствами, до моделирования глобальных атмосферных процессов и тектонических разломов в земной коре. Это же привело к прорыву в инженерных расчетах – конструктор – расчетчик мог видеть как выглядит разработанная им система в динамике!

При этом фантастически вырос круг решаемых с использованием прикладных пакетов задач.

Например, только в области биохимии мои друзья из МГУ в 2007 насчитали около 400 пакетов (несколько лет назад!).

Итак, - **CAE (Computed Aided Engineering)**- программные комплексы для инженерного и естественнонаучного анализа 33

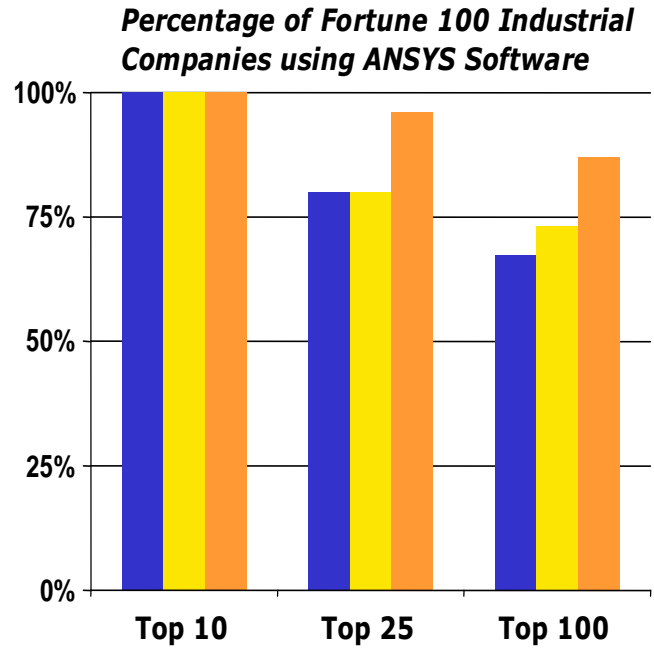
Заметим, что сегодня более принято говорить не о пакетах прикладных программ, а о программных комплексах (или системах). С тем чтобы понять каковы роль и место программных комплексов в сегодняшней практике исследовательских центров и предприятий приведем характерный пример.

Возьмем программный комплекс **ANSYS** выбрав его первые версии на пороге веков.

ANSYS INCORPORATED

Top Companies Use ANSYS

Meeting the needs of a diverse marketplace



- 84 of the top 100 companies
- 24 of the top 25 companies
- All of the Top 10

На рис. 4 динамика его применения 10, 25 и, наконец, 100 ведущими научными центрами и фирмами США в 1998-2000 годах!!!
Комплекс **ANSYS** проник из своей «материнской» области, ядерной энергетики, во все области промышленности, завоевав прочные позиции по всему миру.

Рис 4.

Приведем наиболее востребованные программные системы для инженерного анализа

The logo for ADAMS, featuring the word "ADAMS" in white, bold, sans-serif capital letters on a black rectangular background.The logo for ANSYS, featuring the word "ANSYS" in white, bold, sans-serif capital letters on a black background, with a yellow registered trademark symbol.The logo for ALGOR, featuring the word "ALGOR" in red, italicized, sans-serif capital letters on a blue background, with a red diamond symbol to the left.The logo for CFX, featuring the word "CFX" in a stylized, outlined font with a blue and purple graphic element to the right.The logo for LS-DYNA, featuring a stylized "LS" in teal and "DYNA" in red, with a teal graphic element to the left.The logo for Star-CD, featuring a yellow circle with a black "D" and a red arrow, next to the text "Star-CD" in blue, underlined.

ANSYS - предоставляет одну из наиболее широко используемых и известных программных систем мирового уровня. Главное его предназначение решение задач механики твердого деформируемого тела. В этой области имеются и другие системы из которых упомянем **MSC/NASTRAN**.

CFX, Fluent, STAR-CD. Для решения трехмерных задач механики жидкости и газа наиболее востребованы комплексы- (стационарные и нестационарные течения, ламинарные и турбулентные течения со свободными поверхностями; многофазовые потоки, учет кавитации, химические реакции и др.).

ABAQUS, LS-DYNA, MSC/MARC. Задачи о сильно нелинейных и быстропротекающих процессах в деформируемых средах со сложными реологическими свойствами и для решения пространственных динамических нелинейных задач контактного взаимодействия (например, краш - и дроб - тесты), задачи технологической механики и механики конструкций, содержащих жидкость предмет комплексов

В этих системах заложены и при выходе очередных версий закладываются самые последние достижения механики, математики и вычислительных методов, а версии выходят один – два раза в год.

Безусловно все приведенные рассматриваемые комплексы могут использоваться на самых разных компьютерах, включая и суперкомпьютеры.

Однако, здесь появляется тонкий момент связанный с тем что для суперкомпьютеров такие системы должны использоваться **в многопроцессорном режиме, т.е. иметь многопроцессорные версии.** И, действительно, многие из приведенных систем имеют такие версии. И на этом мы детально остановимся далее.

А сейчас, дабы Вы понимали как организовано современное компьютеризованное производство мы остановимся на том **как компьютерные повлияли на сам процесс производства!**

Итак!

Проблемы организации командной работы над созданием новых изделий и систем в сжатые сроки, очевидно предполагают мощную информационную поддержку этого процесса, содержащего данные об изделии на протяжении всего его жизненного цикла.

Эти проблемы стали привлекать к себе внимание с 80-х годов прошлого столетия.

Для их решения предлагались различные подходы, например, Министерство обороны США предложило методику описания жизненного цикла продукции (конечно в первую очередь с позиций автоматизированной поддержки принятия решений по приобретению изделий и материально-техническому обеспечению (**CALS, Computer-aided Acquisition and Logistics Support**)).

Корпорация IBM предложила концепцию компьютерного интегрированного производства (**CIM, Computer Integrated Manufacturing**).

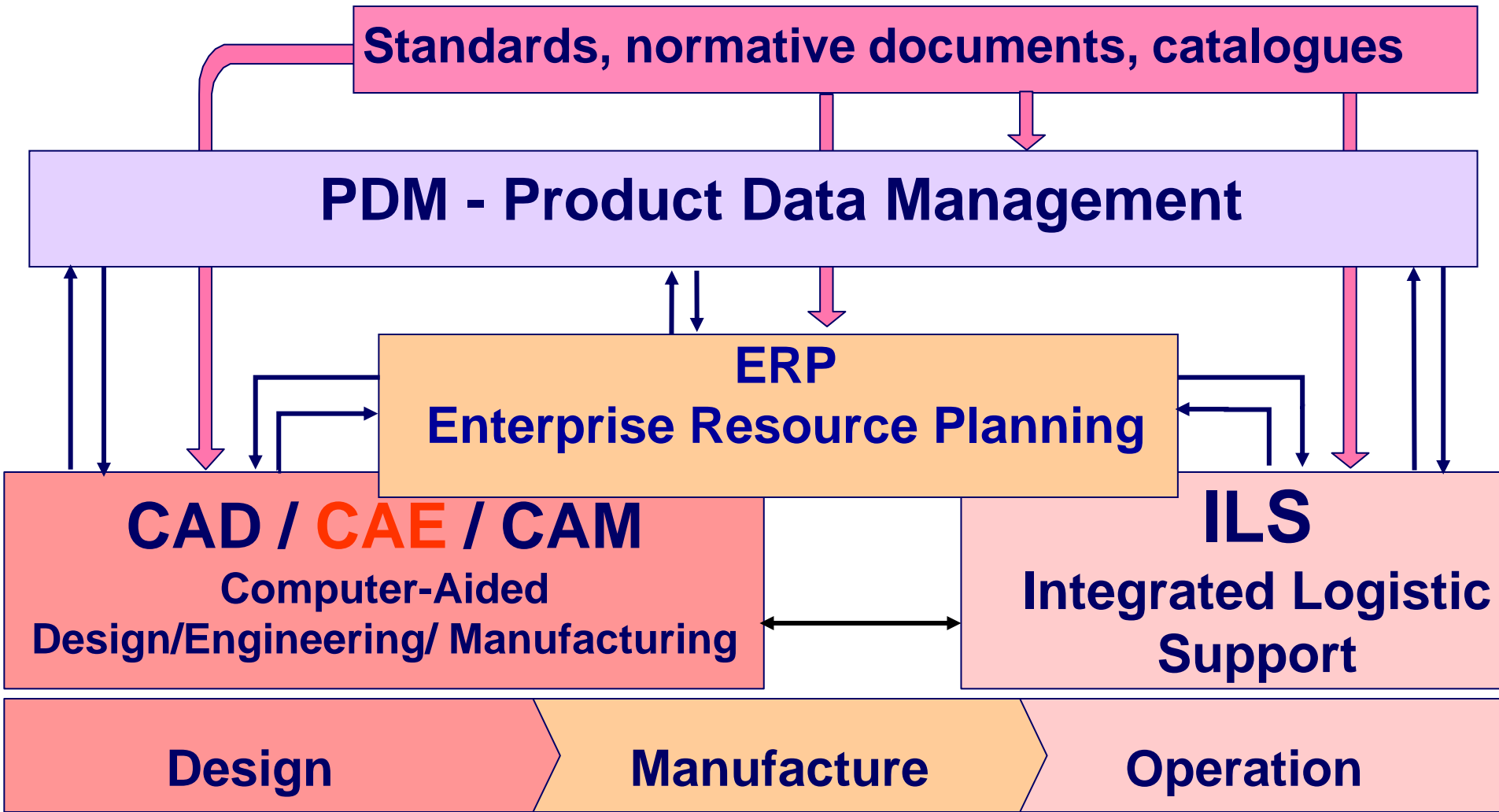
Эти подходы, пожалуй, несколько опередили свое время и не получили очень широкого распространения.

Тем не менее, важно понять важнейшие аспекты идей **CALS** и **CIM** идеологий. Рассмотрим это на примере **CALS**-технологий.



CALS Technologies

(Computer-aided Acquisition and Logistics Support)



Широта постановки проблемы **CALS** – технологии предполагает содержание в них экономических, управленческих и иных задач, которые, конечно, лежат также составляют важные аспекты процессы производства. Но нас, конечно, в первую очередь интересует основа инженерной деятельности, - т.е. инженерный анализ и проектирование.

Итак, - проектирование, инженерный анализ и передача его итогов в производство. Это реализуется наукоемкой триадой - цепочкой **CAD/CAE/CAM** – технологий, где традиционный последовательный подход к разработке изделий заменен принципиально новым, интегрированным подходом, получившим название «параллельное проектирование».



В его основе лежит идея совмещенного во времени проектирования изделия (**CAD-design**), выполнения системы инженерных расчетов (**CAE-engineering**) и, наконец, подготовки производства (**CAM-manufacturing**). Это позволяет использовать проектные данные, начиная с самых ранних стадий разработки изделия, одновременно различными группами специалистов, а затем их итоги передавать в **PDM (Product Data Management)** – блок.

Таким образом, именно эта «триада» - цепочка **CAD/CAE/CAM** – технологий содержит в себе самое наукоемкое, естественнонаучное и инженерное ядро **CALS**– технологий.

Итак, **мы подошли к самому главному**. Именно цепочка **CAD/CAE/CAM** – технологий, из которых наиболее трудными как в освоении, так и в реализации, безусловно, являются **CAE** – технологии, технологии компьютерного инжиниринга, лежит в основе современного инженерного анализа и проектирования. И в реализации их на основе суперкомпьютеров в промышленности, и во «встраивании» таких технологий в процесс подготовки кадров. И именно **CAE** – технологии должны составлять базис современного инженерного образования.

Вопросам **CAE** – технологий, технологии компьютерного инжиниринга мы посвятим значительное место далее, а теперь рассмотрим наиболее важные аспекты современных интегрированных производственных технологий, прямо связанные с суперкомпьютерными технологиями и GRID – средами.

В конце 90- годов прошлого века компания IBM разработала новую концепцию описания жизненного цикла продукции– **PLM (PLM, Product Life Management)**

которая получила значительное распространение, чем **CALS-** и **CIM-** технологии (как программные инструментари). По существу **PLM-** инструменты, - это совокупность программных систем с объединением и эффективным взаимодействием отдельных сегментов автоматизации, образовавшихся в результате внедрения различных программным систем – **CAD/CAM, CAE, PDM, ERP** и других в рамках единого информационного пространства предприятия, в первую очередь для реализации непрерывного (интегрированного) конструкторского, технологического и коммерческого циклов производства продукции «от идеи создания продукта, его разработки, эксплуатации и, наконец, до его утилизации».

Принципиально важно понимать, что основу **PLM-**технологий составляют ядро из множества компьютерных технологий: **CAD/CAM-, CAE- и PDM** и других технологий, благодаря совместному использованию которых традиционный последовательный подход к разработке новых изделий заменен современным интегрированным подходом.

Часть 4. Как компьютеризированное производство развивается сегодня и как это развитие связано с суперкомпьютерными технологиями!?

Сегодня основополагающая концепция «**Simulation Based Design**» интенсивно развивалась силами ведущих фирм-вендоров САЕ-систем и передовых промышленных компаний. Особое значение сегодня приобрели решения задач по проектированию изделий и систем на основе максимально приближенных к реальности междисциплинарных (мультидисциплинарных) постановок задач **MultiDisciplinary Simulation Based Design/ Engineering**. Но! Такие междисциплинарные постановки задач требуют очень больших вычислительных ресурсов, поскольку в их основе постановка **СВЯЗАННЫХ начально краевых или краевых задач математической физики!!!** Например задачи горения, – это связка задач аэрогидродинамики, тепло и массообмена, излучения и, наконец, физико – химической кинетики!

Это породило широкую гамму технологий:

SuperComputer Simulation Based Design (на основе HPC-технологий (High Performance Computing), суперкомпьютеров, высокопроизводительных вычислительных систем (сред именно здесь GRID – среды!) для решения таких задач, выполнения много модельных и много вариантных расчетов), это т.н.

1) **SuperComputer (MultiScale / MultiStage * MultiDisciplinary * MultiTechnology) Simulation Based Design/Engineering**, где реализуется триада: многомасштабность/многостадийность*мультидисциплинарность*

мультитехнологичность

2) **SuperComputer (Material Science * Mechanics) (Multi**3) Simulation Based Design/Engineering** (одновременное проектирование и создание материалов и элементов конструкций из них – гармоничное объединение механики материалов и конструкций на основе **суперкомпьютеров**).

3) **SuperComputer (SmartMat*Mech)*(Multi**3) Simulation and Optimization Based Design / Engineering** (применение Smart-материалов / “умных” материалов, применение разных видов оптимизации параметрической, многомерной, многокритериальной и т.д.), рациональной оптимизации технологических процессов и т.д. опять таки на основе **суперкомпьютеров**).

4) **SuperComputer (SmartMat*Mech)*(Multi**3) Simulation and Optimization Based Product Development** (проектирование, инжиниринг и оптимизацию с расширением до производства продукции и переходом к Virtual Product Development – виртуальной разработке продукции / изделий с использованием **суперкомпьютеров**).

5) **Digital Mock-Up / Digital Manufacturing** («цифровой прототип» – виртуальная, цифровая 3-D модель изделия и всех его компонентов, позволяющая **исключить из процесса разработки** изделия создание дорогостоящих натуральных моделей-прототипов, позволяющая «измерять» и моделировать любые характеристики объекта в любых условиях эксплуатации / «цифровое производство» – как основные компоненты «умных» заводов и фабрик).

Итак. Теперь перейдем к рассмотрению ядра современного инженерного анализа - CAE-систем

CAE, Computer-Aided Engineering – компьютерный инжиниринг, основанный на применении мультидисциплинарных программных CAE-систем. Его фундамент, это, как правило, связанные начально-краевые или краевые задачи математической физики в неклассической постановке на основе интегральных тождеств (нестационарные системы нелинейных дифференциальных уравнений в частных производных или их интегральные аналоги).

Вычислительные подходы построены на **FEA (Finite Element Analysis** – методе конечных элементов) и **CFD (Computed Fluid Dynamics** методы вычислительной гидродинамики) и других современных численных методах.

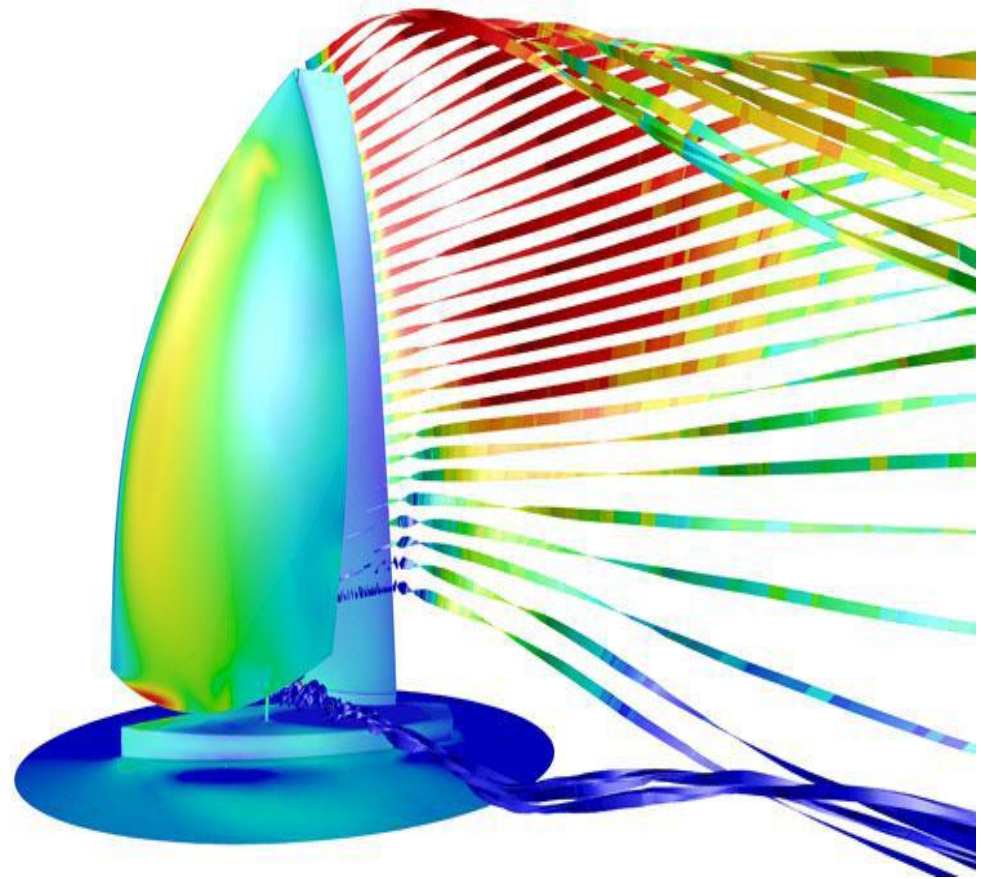
Наиболее передовые CAE-системы описывают поведение машин и систем максимально близко к реальному физическому миру и позволяют конструировать но высоком уровне адекватности реальности сложные конструкции и механических систем на основе **MBD (Multi Body Dynamics)** – технологий.

При этом имеются специализированные системы, например, **MCAE (Mechanical CAE)**, **ECAE (Electrical CAE)**, **AEC (Architecture, Engineering and Construction)** и т.д.

Сегодня рекордами по числу расчетных узлов является

- для **CFD-задач - 10^9 ячеек**, - задача моделирование гидроаэродинамики океанской яхты с использованием разработок ANSYS. В рамках этой работы группа итальянских исследователей и инженеров, работавших с командой Luna Rossa Challenge, участвовавшей в яхтенной регате на кубок Америки 2007, выполнили расчеты в августе 2008 года):

- для **FEA-задач – $5 \cdot 10^8$** уравнений турбомашиностроения с применением CAE-системы NX Nastran от Siemens PLM Software, декабрь 2008 года).



Как САЕ – инструменты работают в многопроцессорном режиме, т.е. как они реализуются на суперкомпьютерах?!

Именно здесь и встает в «полный рост» проблема параллельных вычислительных технологий. Но как или из каких важнейших шагов строится вычислительная процедура для решения задачи на суперкомпьютере? Это шаги:

- 1) *разбиение области на подобласти* для распределения между процессорами (узлами суперЭВМ) и установление связей между подобластями;
- 2) *построение сеток* в каждой из подобластей;
- 3) *выбор вычислительной процедуры* для решения полученной алгебраической системы уравнений.

На этих шагах остановимся подробнее.

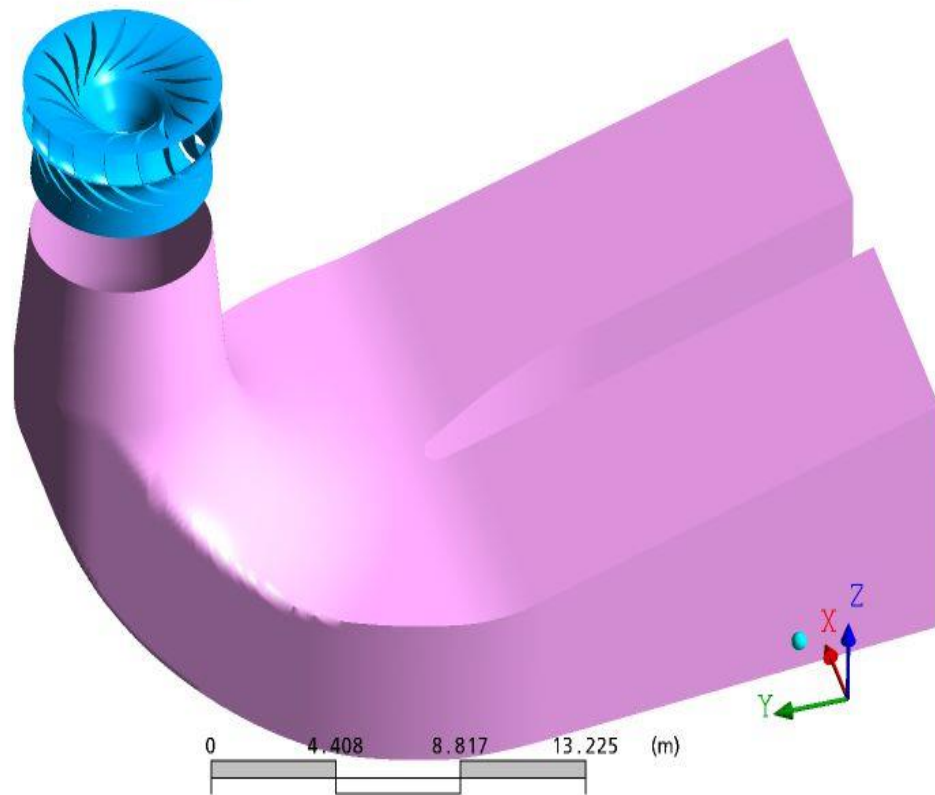
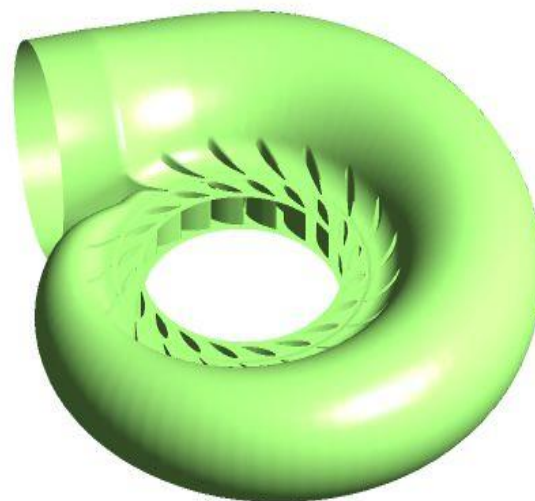
Начнем с примера конкретной задачи о моделировании течения в гидротурбине Саяно - Шушенской ГЭС (работа с ЛМЗ ОАО «Силовые машины»).

Сразу же укажем на наиболее актуальные проблемы:

1. **Устойчивая работа** гидромашины, например, минимизация вибраций.

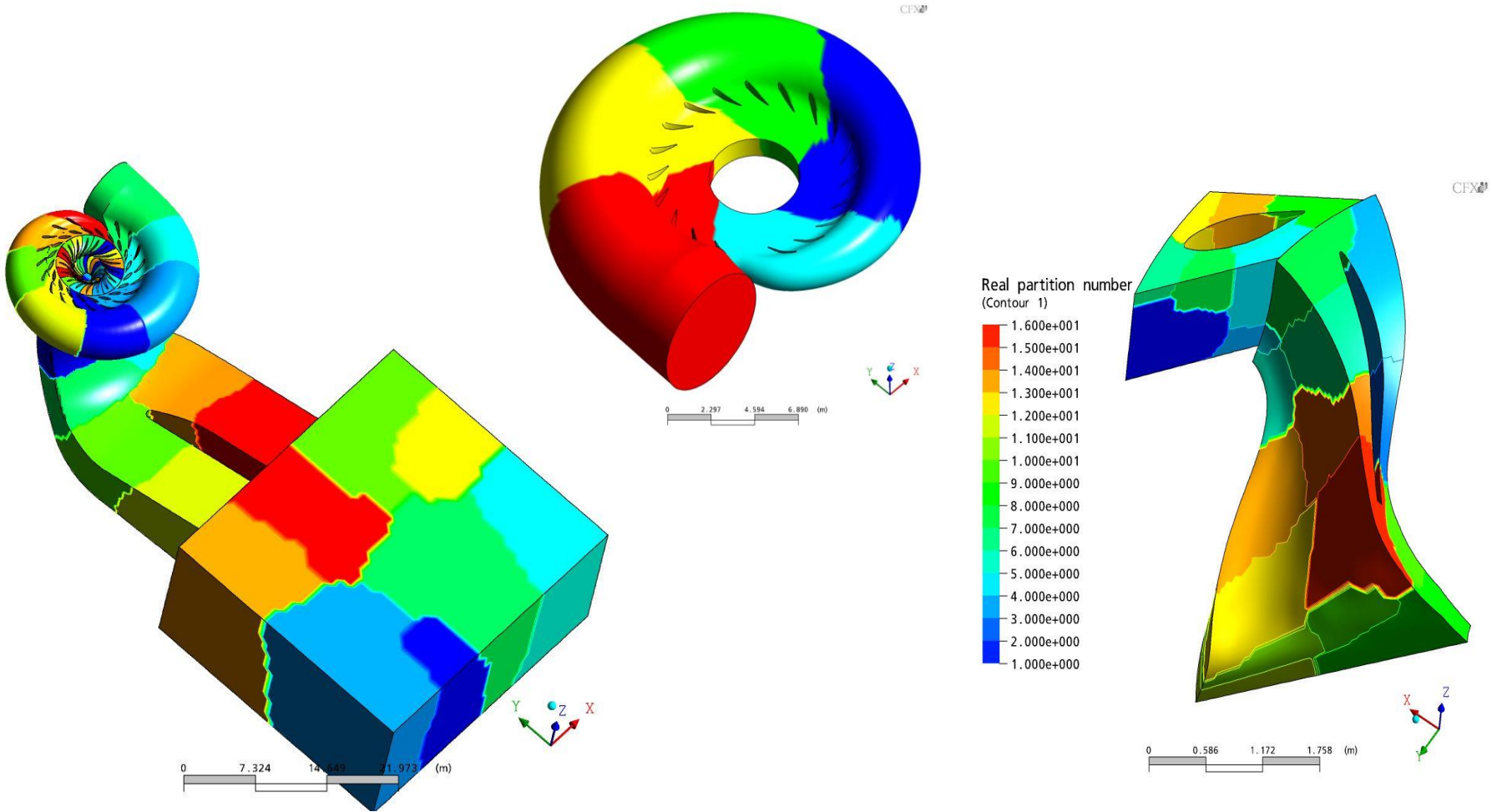
2. **Повышение КПД** отечественных гидротурбин (на мировом рынке борьба идет даже за доли %).

Подчеркнем, решить их можно только математическим моделированием и только на СуперЭВМ (здесь использовалась до 64 процессоров).



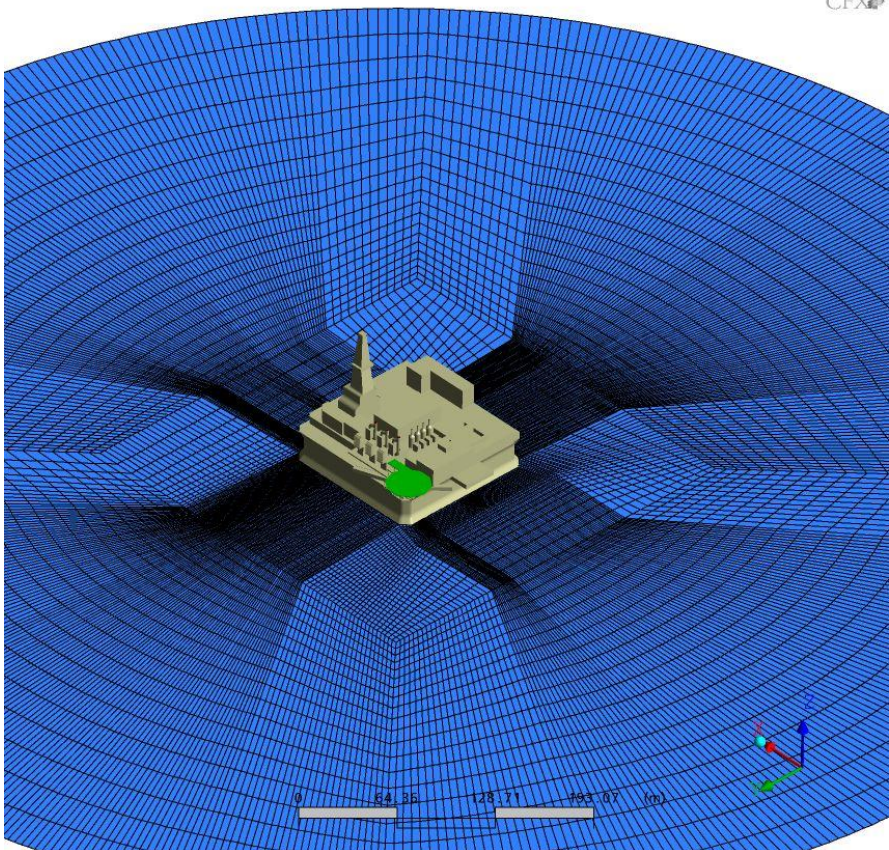
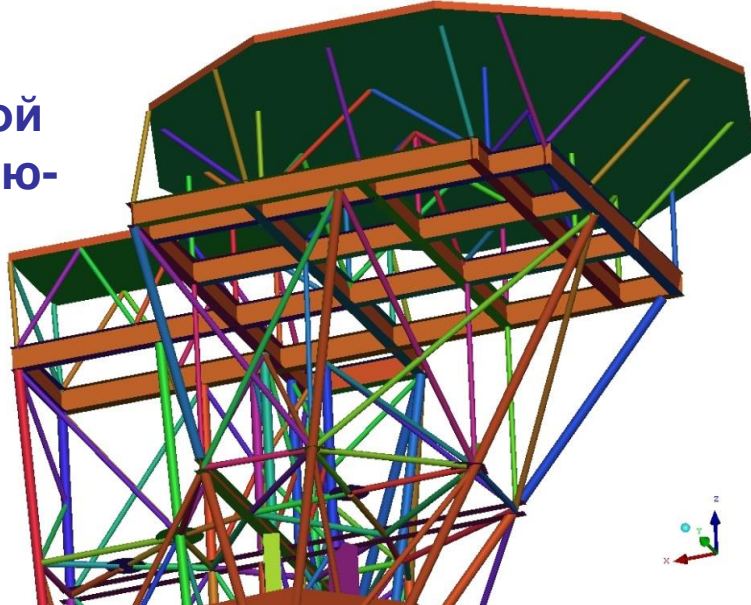
На рисунках показаны разбиения:

- всей области течения,
- улитки и направляющего аппарата;
- области в окрестности лопатки рабочего колеса.

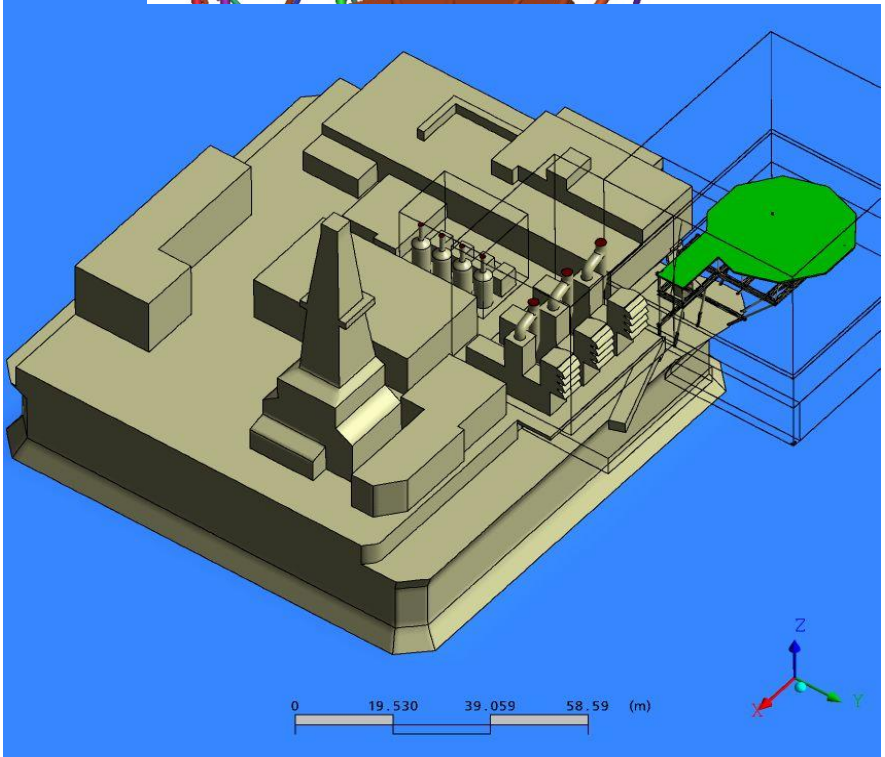


Еще один пример разбиения сложной и масштабной области - моделирование ветровой нагрузки вертолетной площадки нефтедобывающей платформы (работа с ЦКБ МТ «Рубин»).

Разбиение области проводилось на на 24 и 48 подобластей (показаны частично на рис внизу слева)

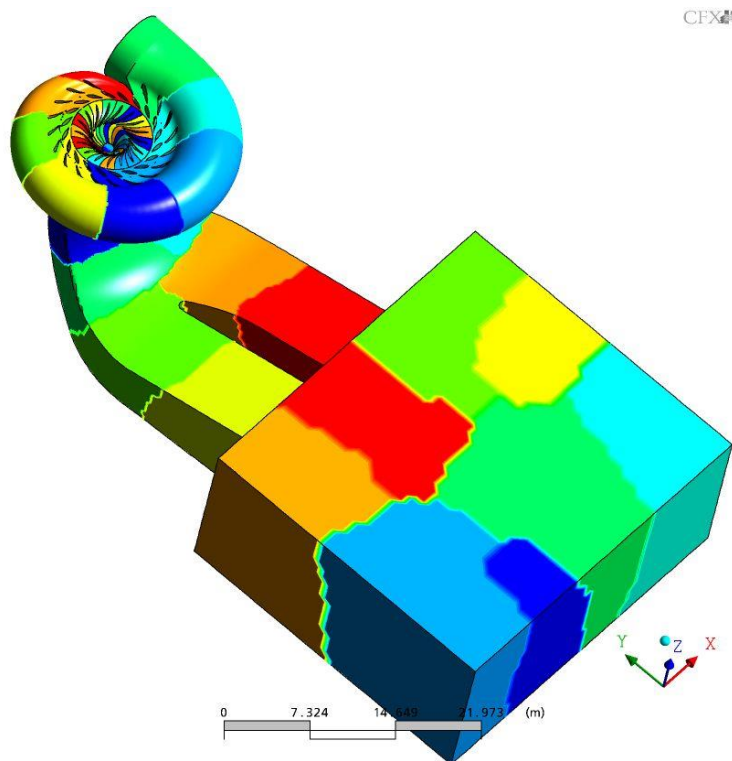


Вся расчетная область

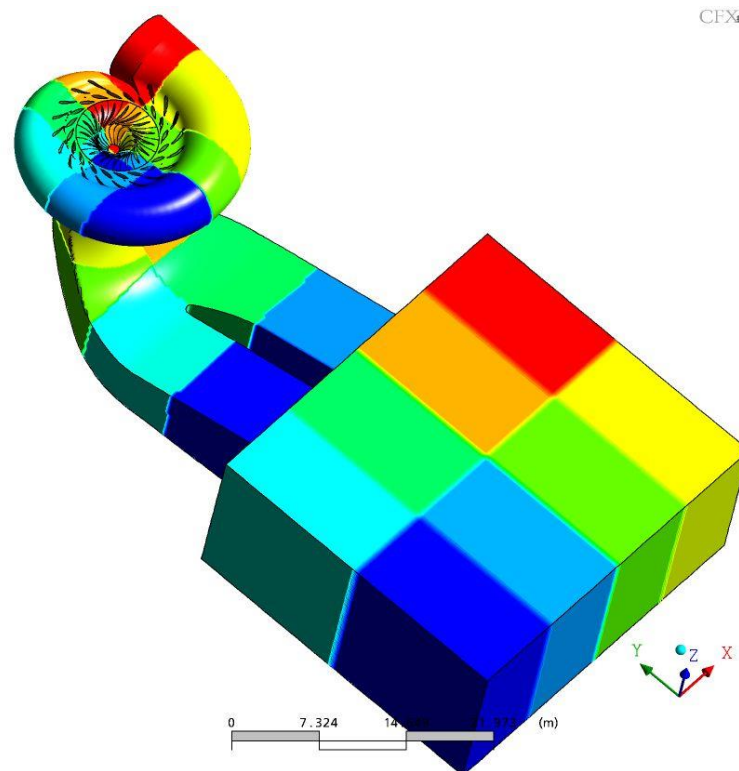


Платформа и площадка

Очевидно, что от характера (способа) разбиение на подобласти зависит ход вычислений. Здесь имеется много различных Технологий. На рисунках разбиение всей расчётной области проведено 2 методами (разбиение каждой отдельной области для большей наглядности приведено на 8 частей



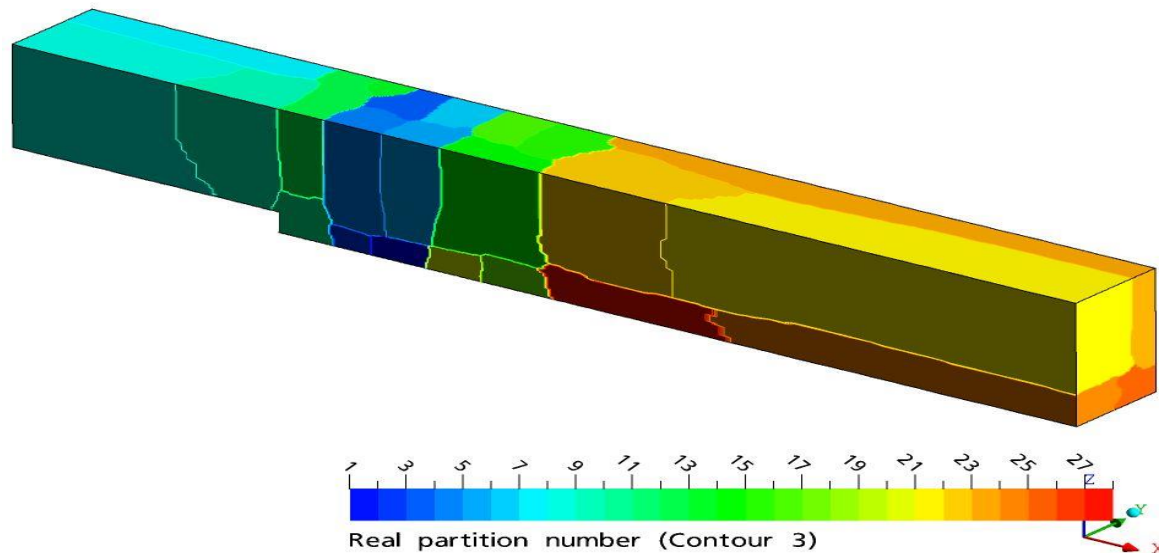
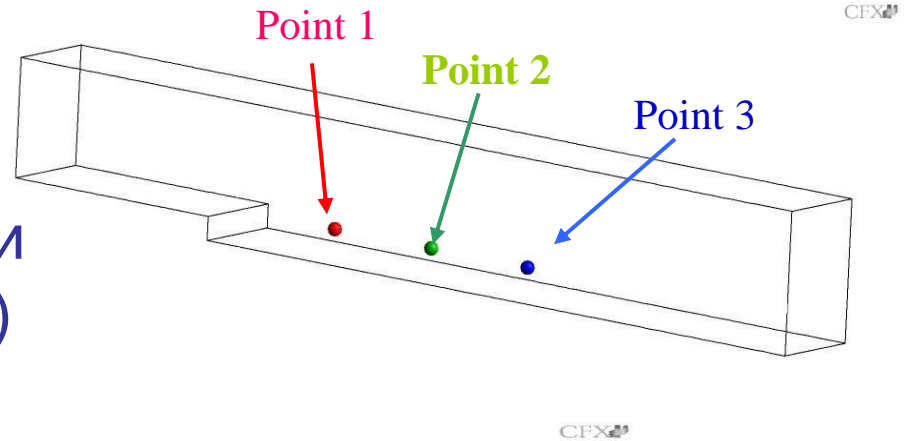
Разбиение расчётной области -
MeTiS-алгоритм



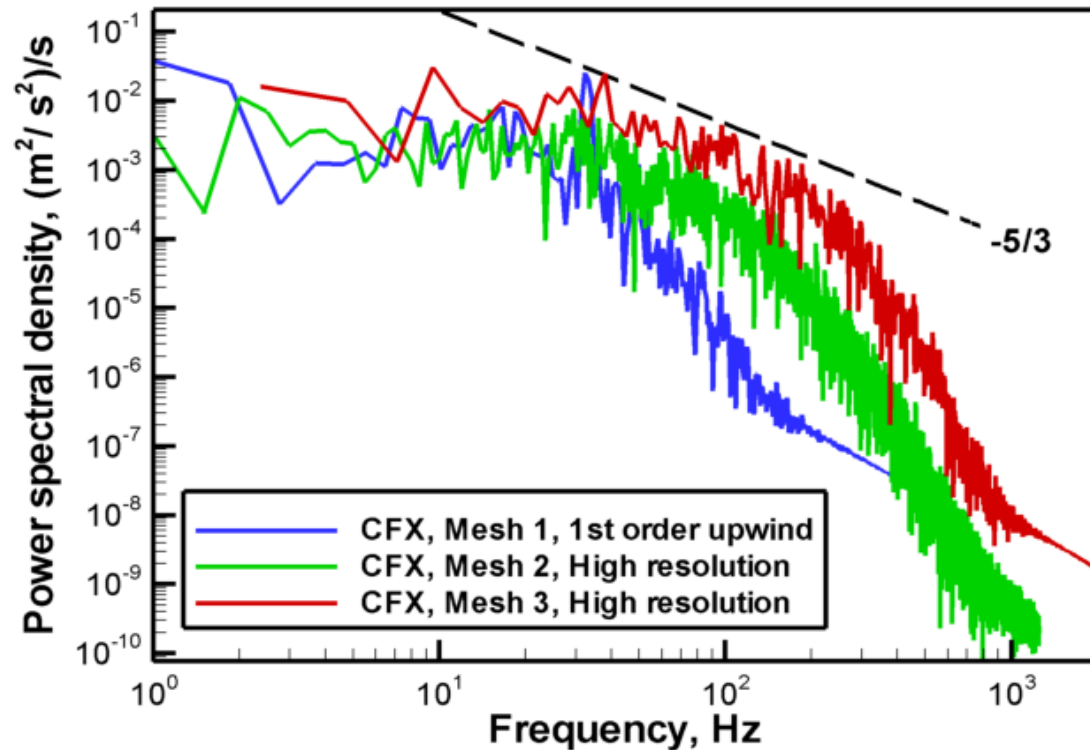
Рекурсивное
покоординатное
половинное деление

Для анализа эффективности моделей турбулентности часто рассматривают течение в так называемом «обратном уступе». На рисунке приведена схема обратного уступа с указанием характерных точек для измерения параметров течения.

На нижнем рисунке приведено характерное разбиение расчётной области (для случая 28 процессоров) на основе METIS-алгоритма.

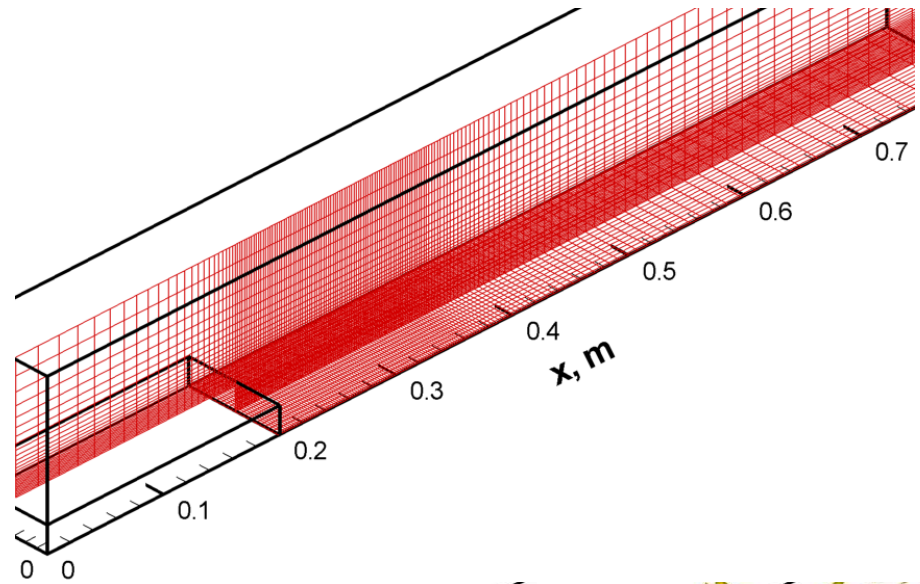
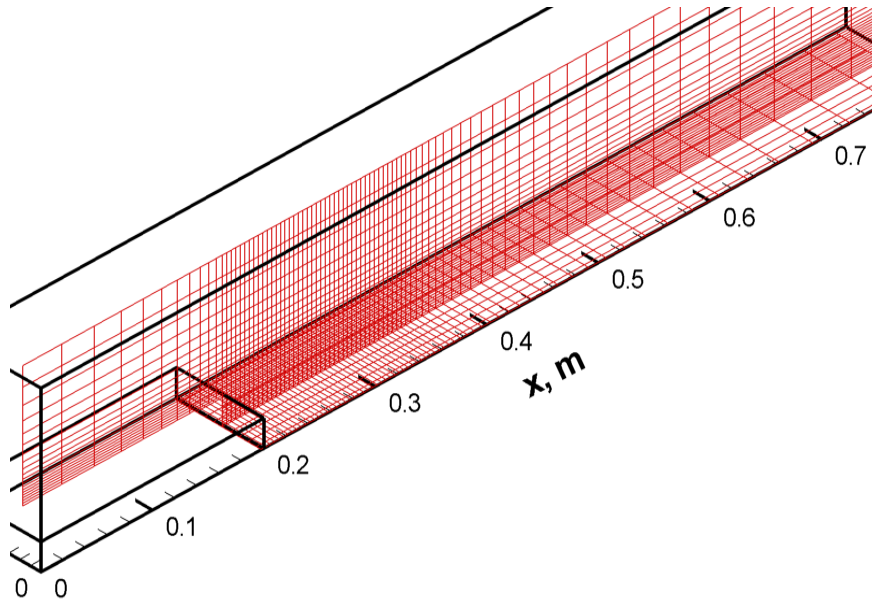


Теперь кратко остановиться на вопросах построения сеток. Очевидно, что влияние той или иной сетки на результаты вычислений весьма велико. Продемонстрируем это на «обратном уступе». На рис. показаны результаты расчета на трех различных сетках на энергетический спектр пульсаций скорости в турбулентном отрывном течении.



При тестировании LES модели турбулентности использовался комплекс **Ansys CFX 11.0**.

Приведем две расчётные сетки в данной задаче - Mesh 1, слева и Mesh 2, справа:

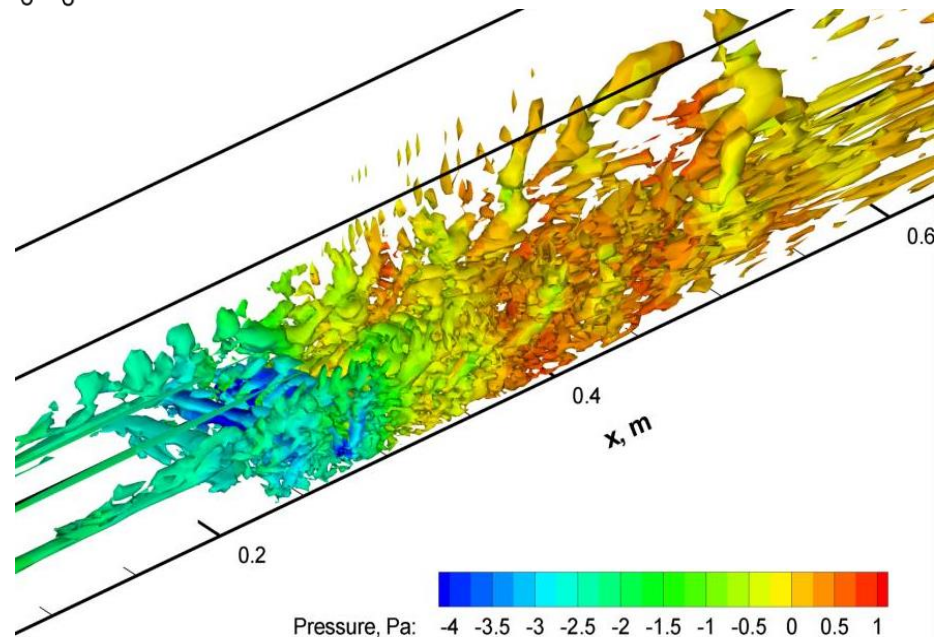


Что считаем!

Справа - мгновенное расчётное поле течения.

Приведены изоповерхности инварианта $I = S^2 - \Omega^2$ (S – модуль тензора скорости деформации, а Ω – модуль тензора завихрённости)

Но вычислителям понятно, что такие сетки – «цветочки»!

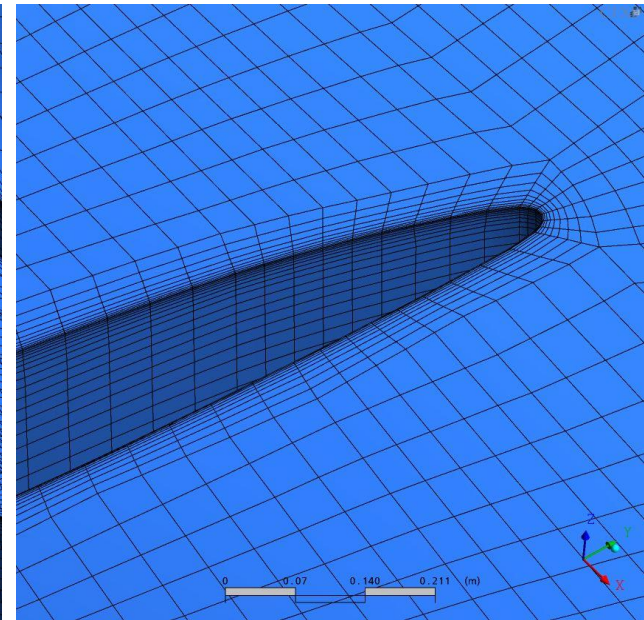
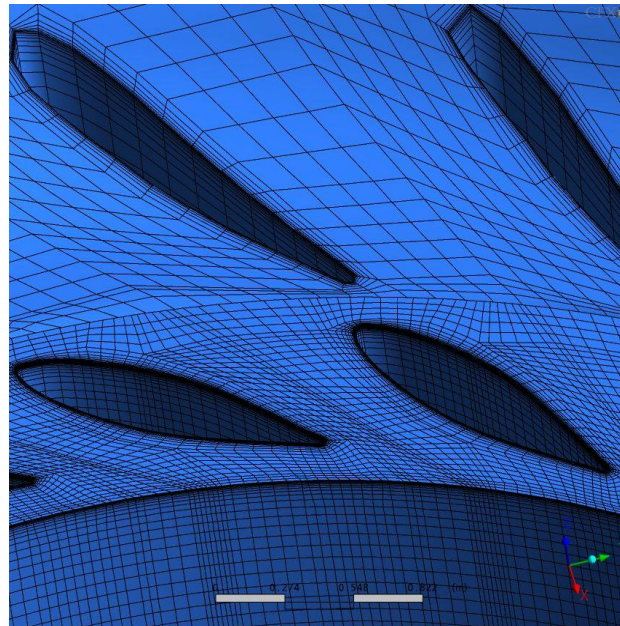
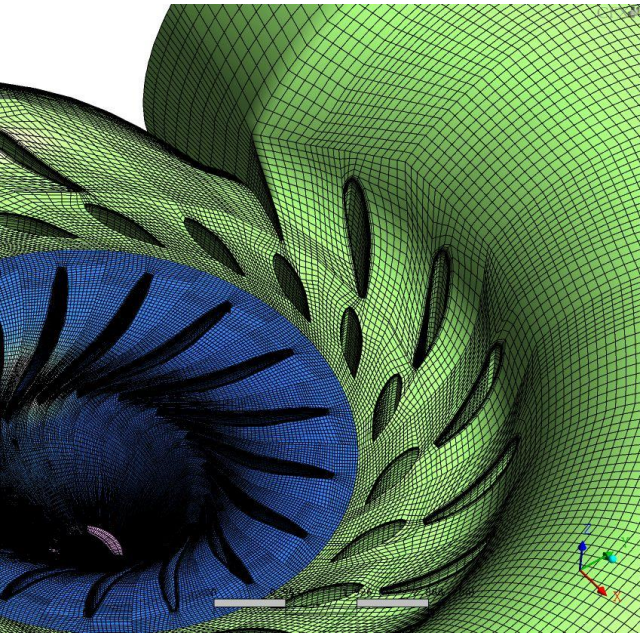


Вообще, задачи построения «хороших» сеток сложны.

Для их решения, сегодня, создано множество

сеткопостроителей. На рисунках приведены характерные сетки построенные для расчета проточных частей гидромашин.

Сетки построены в программных комплексах **ANSYS ICEM CFD** и **ANSYS Blade Gen** (2006 г.).



Вполне очевидно, что примеров построения сеток можно привести множество.

Важнейшим моментом, безусловно является численная процедура, для решения получающихся, как правило, алгебраических нелинейных систем.

Принципиально методы решения больших систем, как мы знаем, делятся на прямые и итерационные. После бурного развития итерационных методов в 50-70 гг прошлого века, исследователи обратились к прямым методам, которые как выяснилось м.б. более эффективны чем итерационные.

В современных программных комплексах представлены как те, так и другие. При этом имеются возможности их использования с разной точностью и в самых разных вариантах (модификациях).

Этими общими соображениями и ограничимся.

Перейдем к такому важнейшему понятию суперкомпьютерных технологий как масштабируемость.

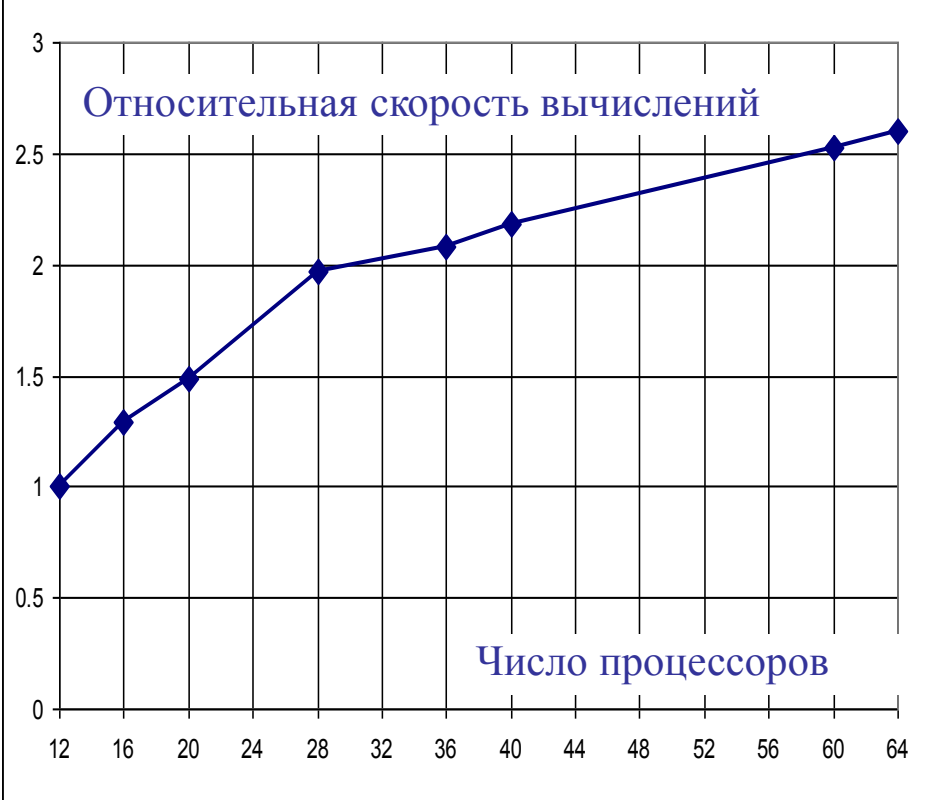
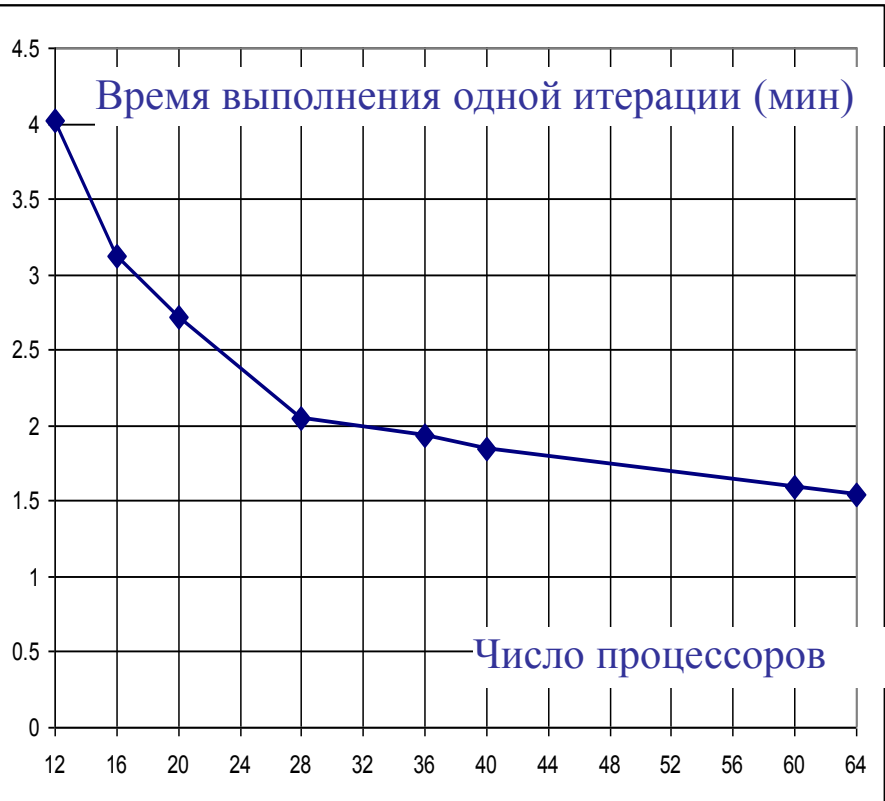
При этом нелишне уточнить это понятие.

Начнем с того, что, вообще говоря, само это понятие может рассматриваться в двух аспектах.

В первом, под масштабируемостью понимают возможность наращивания ресурсов вычислительной системы, то есть числа процессоров, объемов оперативной и внешней памяти и так далее, «делая» вычислительную систему более мощной или, как и принято говорить, масштабируемой. Т.о., здесь **масштабируемость обеспечивается архитектурой** (или конструкцией) компьютера, и соответствующими средствами системного программного обеспечения (*масштабируемость вычислительной системы*).

Во втором аспекте под масштабируемостью будем понимать распределение ресурсов компьютера для решения задачи на уровне параллельной версии программной системы или на уровне написанной нами прикладной «параллельной» программы, то есть на уровне их запросов к вычислительной системе. Т.о., **здесь технология распараллеливания реализуется в вычислительной процедуре**, на основе которой построены алгоритмы программной системы или некоторой программы (*масштабируемость задачи*). Эффективность их реализации, то есть лучшая масштабируемость, выражается в близости к линейной зависимости кривой, характеризующей уменьшение времени решения задачи с увеличением её разбиения на подзадачи, «декомпозиция» задачи о которой говорили ранее. Приведем примеры.

На рисунке слева приведено время выполнения одной итерации в зависимости от числа разбиений области (или от числа задействованных в расчёте процессоров) при расчетах турбомашины



Справа приведен график относительной скорости вычислений. Как видим время выполнения одной итерации как и скорость вычислений с ростом числа процессоров, начиная с какого то момента начинают расти слабо с ростом числа задействованных в расчётах процессоров. А лучший результат будет при линейном росте!

И еще одно важнейшее обстоятельство для инженерных и естественнонаучных расчетов. Это проблема **визуализации результатов вычислений! Очевидна крайняя важность этой проблемы в инженерном деле!**

Переходя к примерам решения ряда инженерных задач центральное место как раз и уделим проблеме визуализации результатов.