
FlowVision – что внутри



Технический директор ТЕСИС, к.ф.-м.н.
Андрей Александрович Аксенов

FlowVision - история

- Первые работы (для НПО Энергии, руководитель – космонавт А.А. Серебров) – исследование гидродинамики установок для производства белка в условиях невесомости. ¹⁾
- Начало разработки FlowVision 1.0 – ИАП РАН, 1991г. Заказчики – чистые комнаты, очистка дымовых газов, горелки для ТЭЦ, РКК Энергия – «Морской Старт».
- 1999 года – коммерциализация разработки в компании ТЕСИС. ²⁾
- Ядро команды - научная школа академиков
О.М. Белоцерковского и Э.Е. Сола
- Основные разработчики - выпускники
МФТИ, МГУ, МАИ, МГТУ им. Баумана



¹⁾ Aksenov A.A., Gudzovsky A.V., Serebrov A.A. Electrohydrodynamic instability of fluid jet in microgravity// Proc. of 5th Int. Symp. on Computational Fluid Dynamics (ISCFD), Aug. 31 - Sept. 3 1993, Sendai, Japan. Japan Society of Computational Fluid Dynamics, Vol. 1, P. 19-24.

²⁾ A. Aksenov, A. Dyadkin, V. Pokhilko. Overcoming of barrier between CAD and CFD by modified finite volume method. Proc. 1998 ASME Pressure Vessels and Piping Division Conference, San Diego, ASME PVP-Vol. 377-1

FlowVision – аттестация в НТЦ ЯРБ

- ✓ Аттестован НТЦ ЯРБ для расчетного обеспечения безопасности ядерных реакторов
- ✓ Три года валидаций на сложнейших тестах – теплообмен, вторичные течения, сопротивления



Решаемые уравнения = математические модели

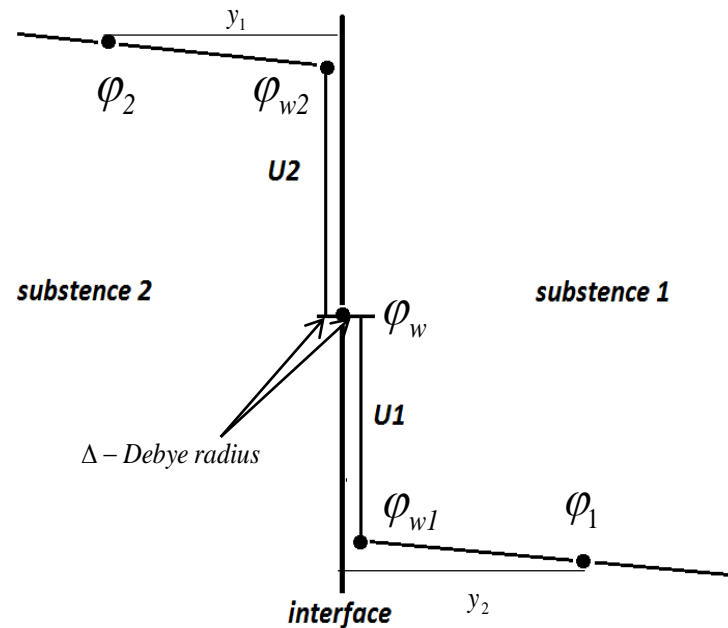
- Уравнения Навье-Стокса
- Уравнение Дарси
- VOF – движение границы раздела несмешивающихся жидкостей
- Модели турбулентности RANS
 - ✓ SA
 - ✓ K-E высокорейнольдсовая
 - ✓ K-E низкорейнольдсовая
 - ✓ SST
- Уравнения переноса дисперсной фазы (Эйлеров подход, спектр по диаметрам частиц)
- Массоперенос + химия
- Массоунос (абляция) с поверхности
- Уравнения Максвелла (стационарные)
- Модель зазора
- Радиационный перенос – тонкий оптический слой, P1, спектральная МДО

Методы решения

- Конечно-объемная сетка с локальной адаптацией
- Второй порядок точности, все-маховый метод расщепления по физическим переменным
- Решение алгебраических уравнений методами AMG (селективный и агрегативный) и неполным GMRES
- Подвижные тела – 6 степеней свободы, движение под действием гидродинамических и пользовательских сил
- Скользящие поверхности
 - «честная»
 - Freezing rotor
 - Сектор, нестационарный
- Решение задач взаимодействия жидкости и конструкций в режиме полного совпадения границы КЭ сетки с сеткой FlowVison, либо через CFD-поверхность
 - Явный каплинг (Абакус, АПМ, Фидесис, Настрэн), протоколы CSE и DC
 - Неявный каплинг (Абакус)
- Подключение собственных программ через API FlowVision и через протоколы CSE, DC, MBC

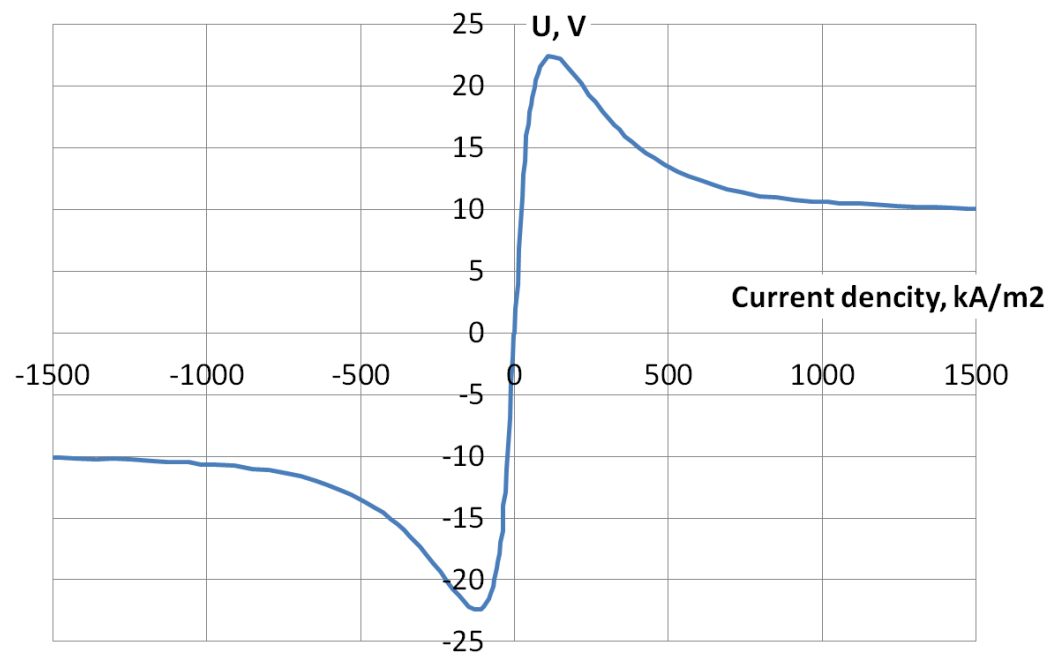
Наши результаты в 2020 отчетном году

Перепад напряжения на границе металл-плазма



На границе раздела металл-ионизированный газ возникает скачок напряжения толщиной в радиус Дебая, который зависит от тока, проходящего по нормали к границе

Вольт-амперная характеристика



Модель электрогидродинамических течений

**Полные
нестационарные
уравнения Навье-
Стокса**

$$\frac{\partial \rho \mathbf{V}}{\partial t} + \nabla(\rho \mathbf{V} \otimes \mathbf{V}) = -\nabla p + \nabla \cdot \hat{\mathbf{t}} + \rho \mathbf{g} + \mathbf{j} \times \mathbf{B}$$

**Уравнение
неразрывности**

$$\frac{\partial \rho}{\partial t} + \nabla(\rho \mathbf{V}) = 0$$

**Уравнение
энергии**

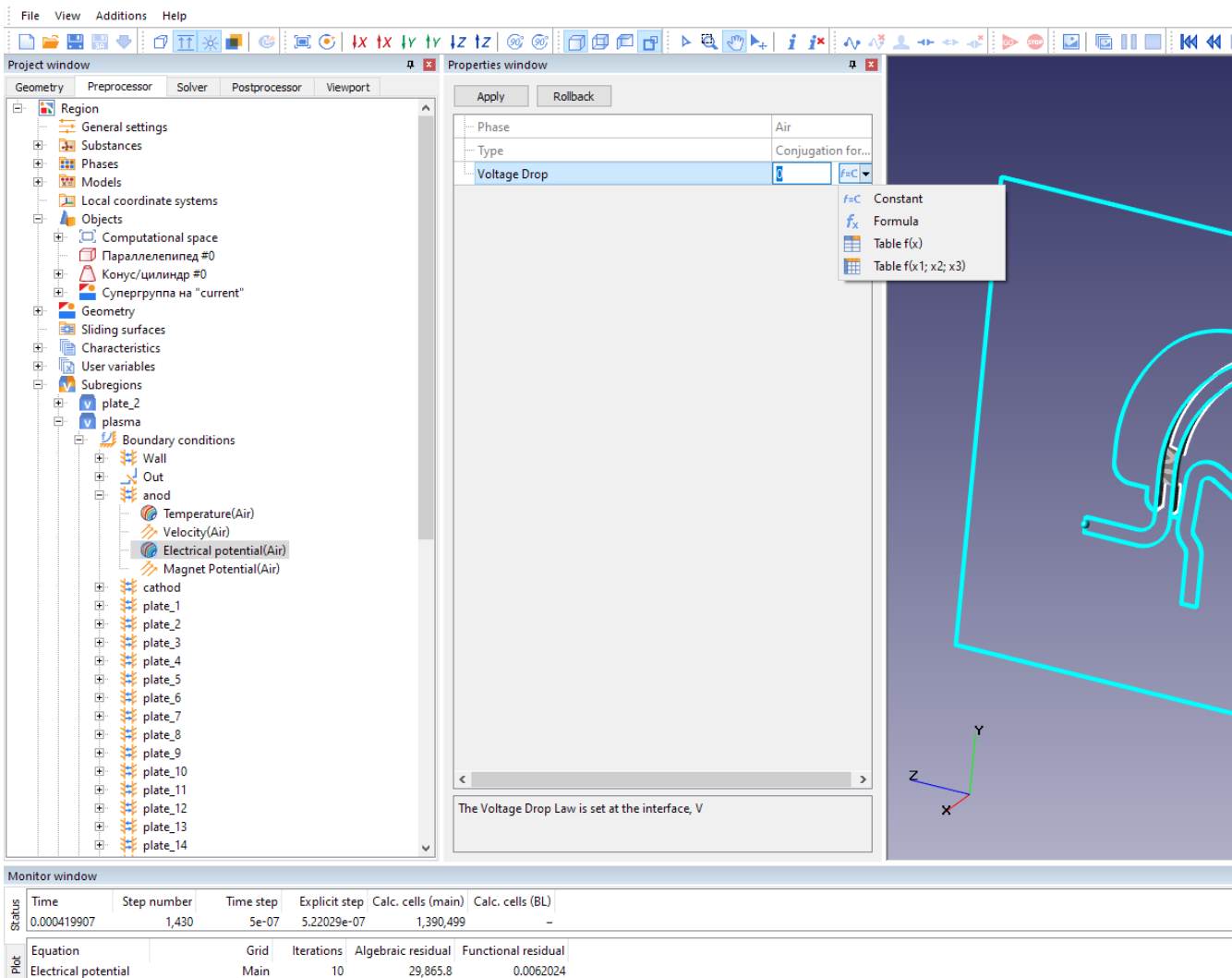
$$\frac{\partial(\rho H)}{\partial t} + \nabla(\rho \mathbf{V} H) = \frac{\partial p}{\partial t} + \rho \mathbf{V} \cdot \mathbf{g} - \nabla \cdot \mathbf{J}_q + \nabla \cdot (\hat{\mathbf{t}} \cdot \mathbf{V}) + j^2 / \sigma \quad H = h + \mathbf{V}^2 / 2$$

Стационарные уравнения Максвелла

$$\nabla \cdot \mathbf{j} = 0 \quad \mathbf{j} = \sigma \mathbf{E} + \mathbf{V} \times \mathbf{B} \quad \mathbf{E} = -\nabla \varphi \quad \nabla \cdot \sigma \nabla \varphi = \nabla \sigma (\mathbf{V} \times \nabla \times \mathbf{A})$$

$$\nabla \cdot \mathbf{B} = 0 \quad \nabla \times \mathbf{B} = \mu_a \cdot \mathbf{j} \quad \mathbf{B} = \nabla \times \mathbf{A} \quad \nabla \times (\nabla \times \mathbf{A}) = \mu_a \cdot (\sigma (\nabla \varphi + \mathbf{V} \times \nabla \times \mathbf{A}))$$

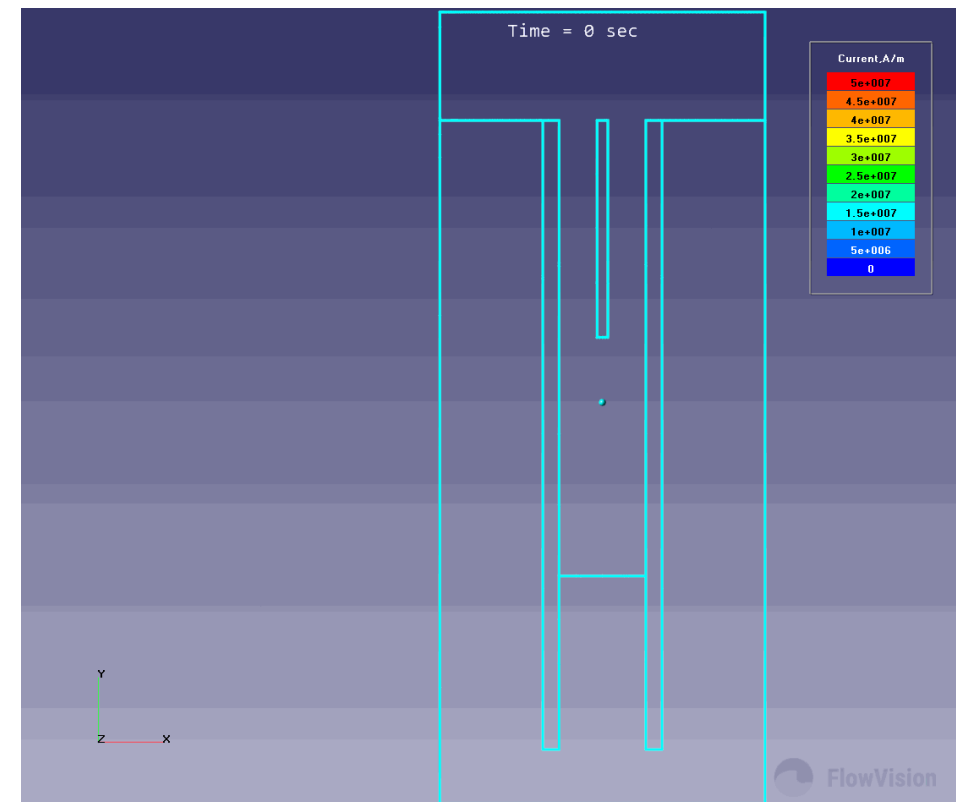
Задание перепада напряжения на границе металл-плазма



The screenshot shows the FlowVision software interface. The 'Properties window' is open for a 'Voltage Drop' boundary condition. The 'Voltage Drop' is set to 'f=C'. A dropdown menu is open showing options: Constant, Formula, Table f(x), and Table f(x1; x2; x3). The 'Monitor window' at the bottom shows simulation progress.

Status	Time	Step number	Time step	Explicit step	Calc. cells (main)	Calc. cells (BL)
	0.000419907	1,430	5e-07	5.22029e-07	1,390,499	-

Plot	Equation	Grid	Iterations	Algebraic residual	Functional residual
	Electrical potential	Main	10	29,865.8	0.0062024



Моделирование в FlowVision переноса излучения методом дробных ординат

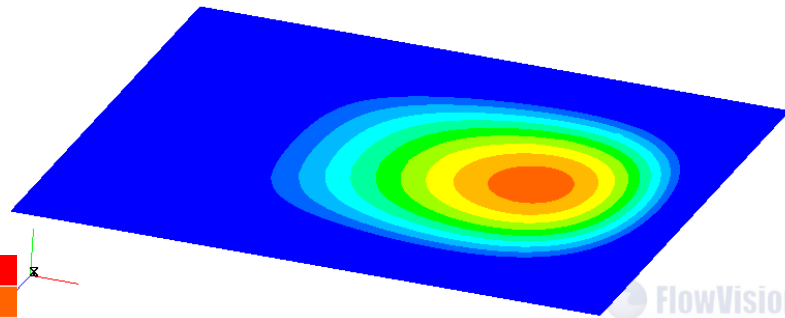
- Реализуется как набор L полос спектра.
- Каждая полоса $\Delta\lambda_n$ определяется диапазоном длин волн $[\lambda_{n-1}, \lambda_n]$.
- Внутри каждой полосы спектральная интенсивность излучения не зависит от длины волны, спектральные коэффициенты поглощения и рассеяния есть усредненные по полосе спектра.
- Полоса спектра излучается из ячейки по N направлениям, N задает пользователь.
- Модель позволяет решать задачи с затенением и перераспределением энергии между частотами излучения

Пример: нагрев источником плоскости

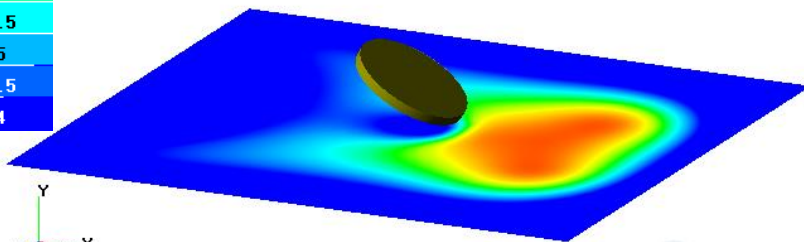
Источник тепла



Расчет без затенения



Расчет с затенением



Распределение температуры

Количество телесных углов - 32

Параметры среды:

$\alpha = 0.01$ – коэффициент поглощения

$\beta = 0$ – коэффициент рассеяния

$n = 1$ – коэффициент преломления

$Q = 10^6$ Вт/м³ – источник тепла

Граничные условия.

поверхность диска:

по температуре – нулевой градиент,

по излучению - непрозрачная стенка с

зеркальным отражением

Граница расчетной области:

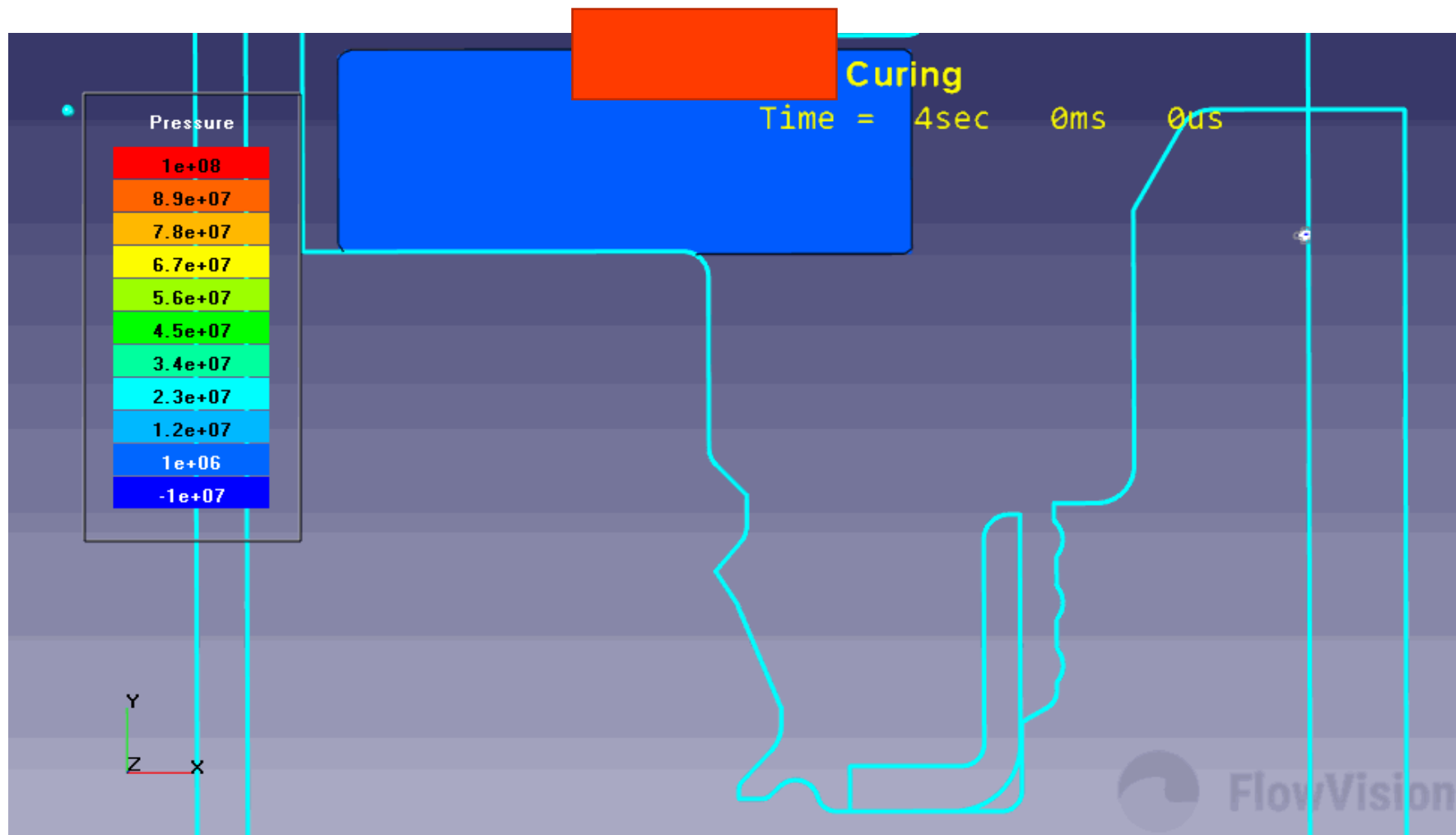
по температуре - внешний теплообмен, $\alpha_{ext} = 10$

по излучению - непрозрачная стенка, $\alpha_w = 1$.

Моделирование очень вязких течений

- При числах $Re \ll 1$ (например $Re = 10^{-5}$) методы расщепления для решения уравнений Навье-Стокса не работают .
- Для этого FlowVision имеет специальный режим работы решателя «очень вязкое течение»
- Можете учесть сжимаемость среды и термическое расширение
- Задачи:
 - Формование резины и эластомеров – с учетом вулканизации!
 - Движение масла в узких каналах уплотнителей (если не используется модель зазора)

Пример «очень вязкие течения» - формование резины

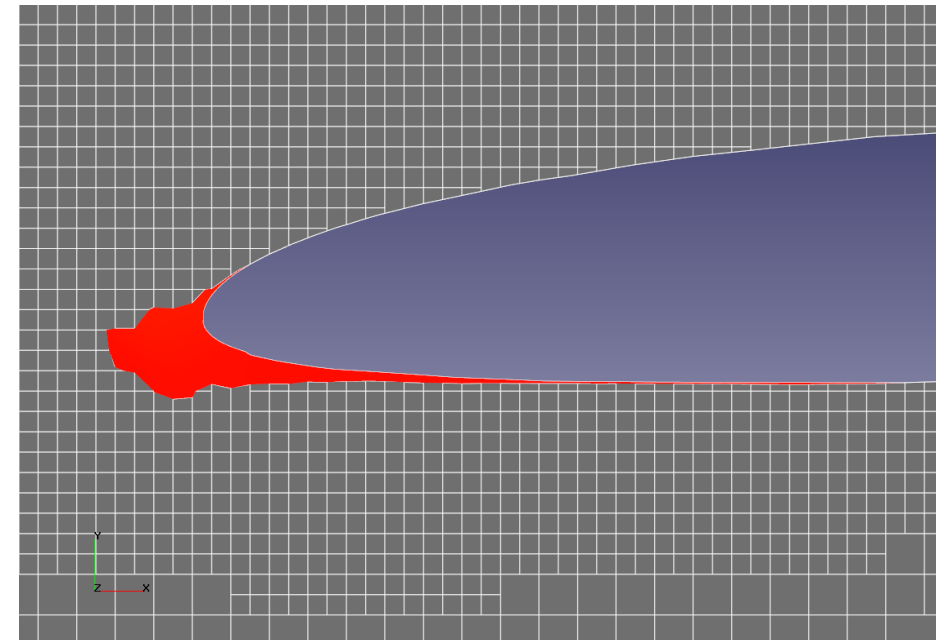
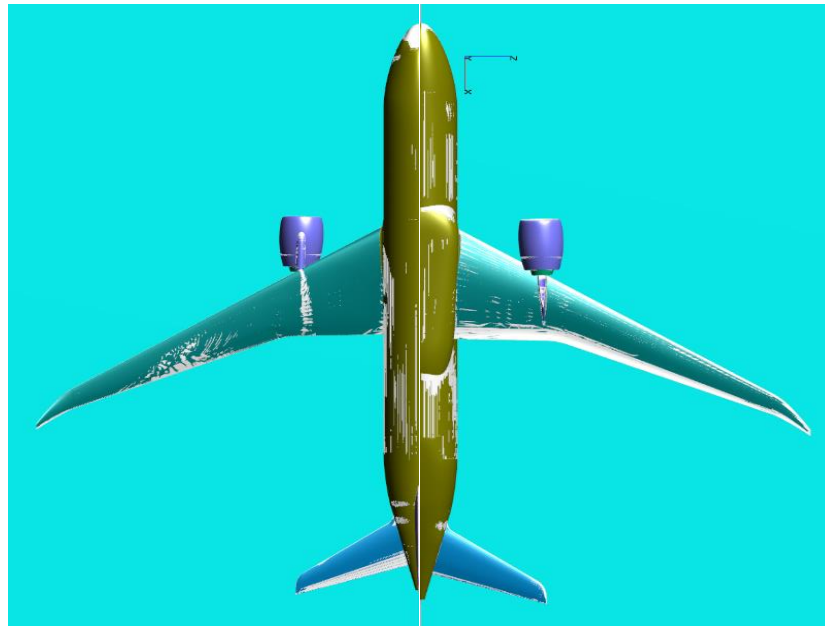


Модель обледенения

- Модель «сухой лед» и модель «лед + пленка воды»
- Лед учитывается с помощью модели VOF (расширение модели FlowVision)
- Перенос тепла во льду, учет нагрева аэродинамических поверхностей самолета
- Испарение пленки воды и возгонка льда



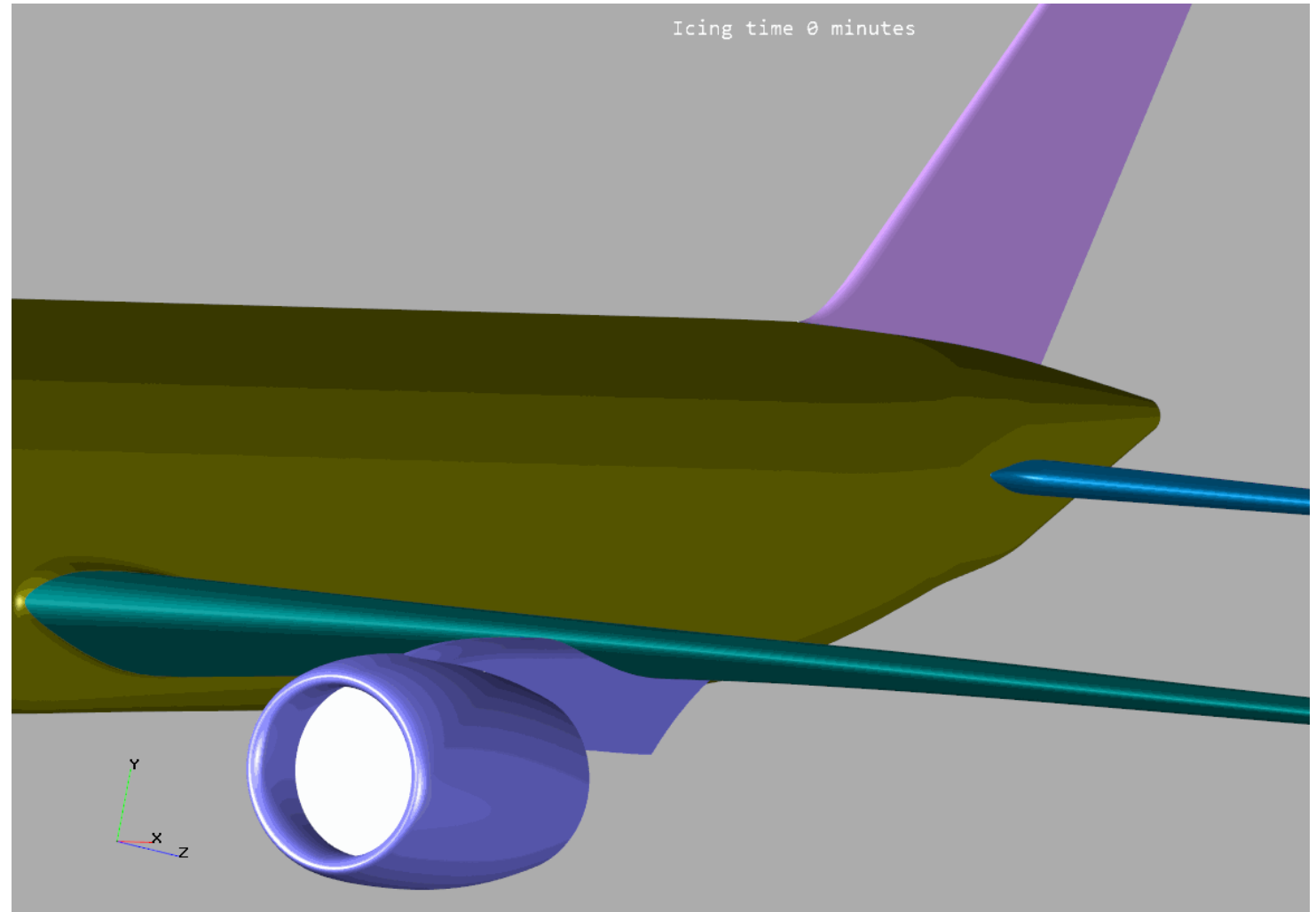
Время полета 22 мин



Ускорение обледенения

У каждого процесса своя временная шкала:

- Воздух : секунды
- Пленка / лед : минуты

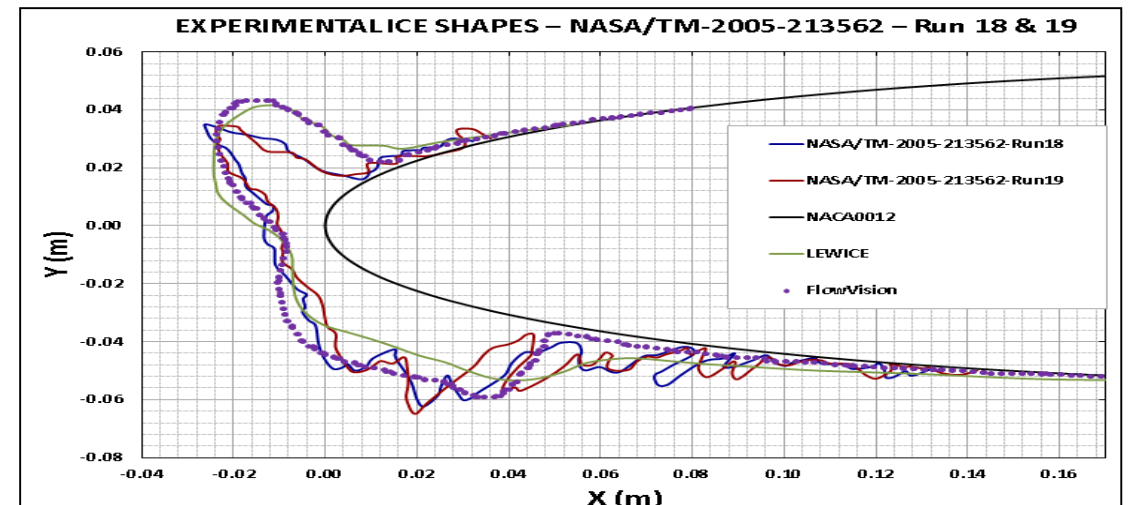
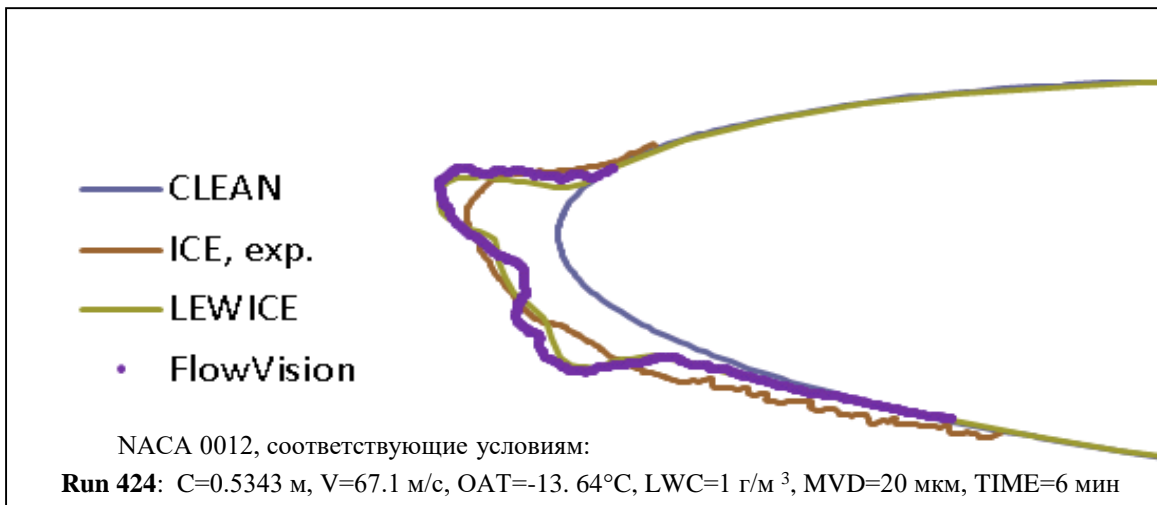
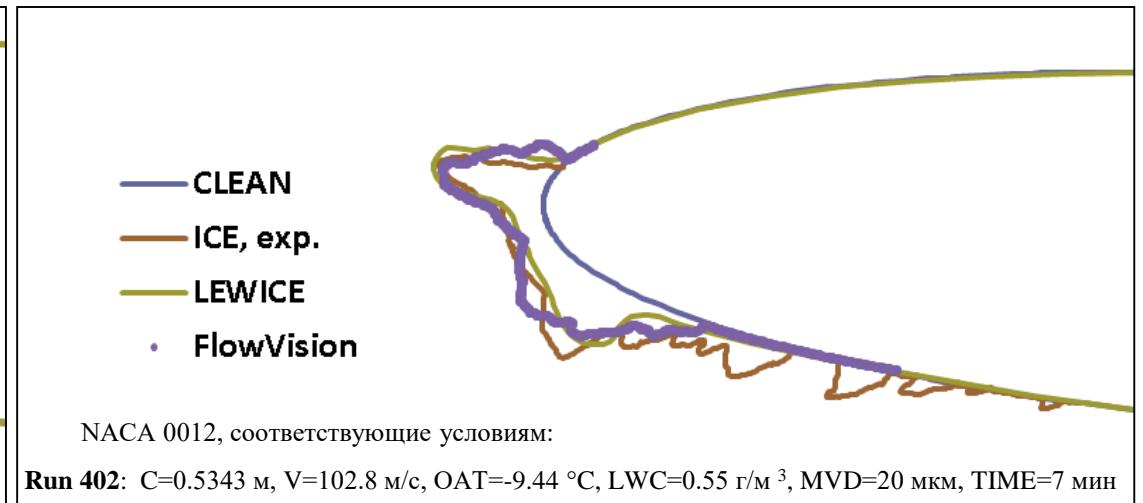
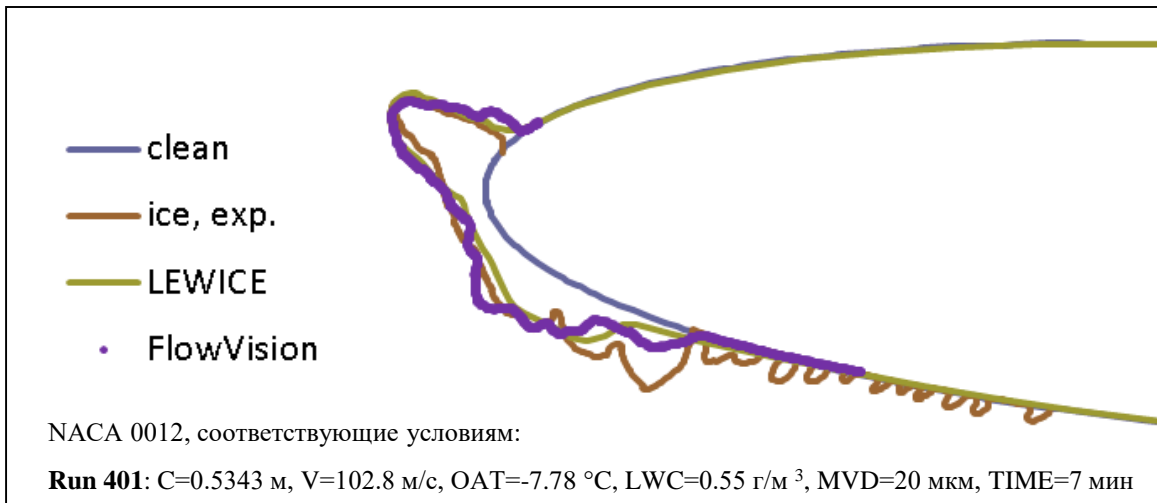


Окно свойств

Применить Откатить

Численный метод	(Разностная сх...
Алгебр. решатель	(Тип решателя...
Расчет нагрузок	(Касательные ...
МГД параметры	(Количество ит...
Многофазность C	(Фаза консерва...
Фаза консервативна	Нет
Релаксация	100000
CFL VOF	0.5
Учет в шаге по времени	Нет
VOF-частицы	Нет
Уровень VOF	1e-05
Использовать консервативную скорость	Нет
Многофазность Д	(Разностная сх...

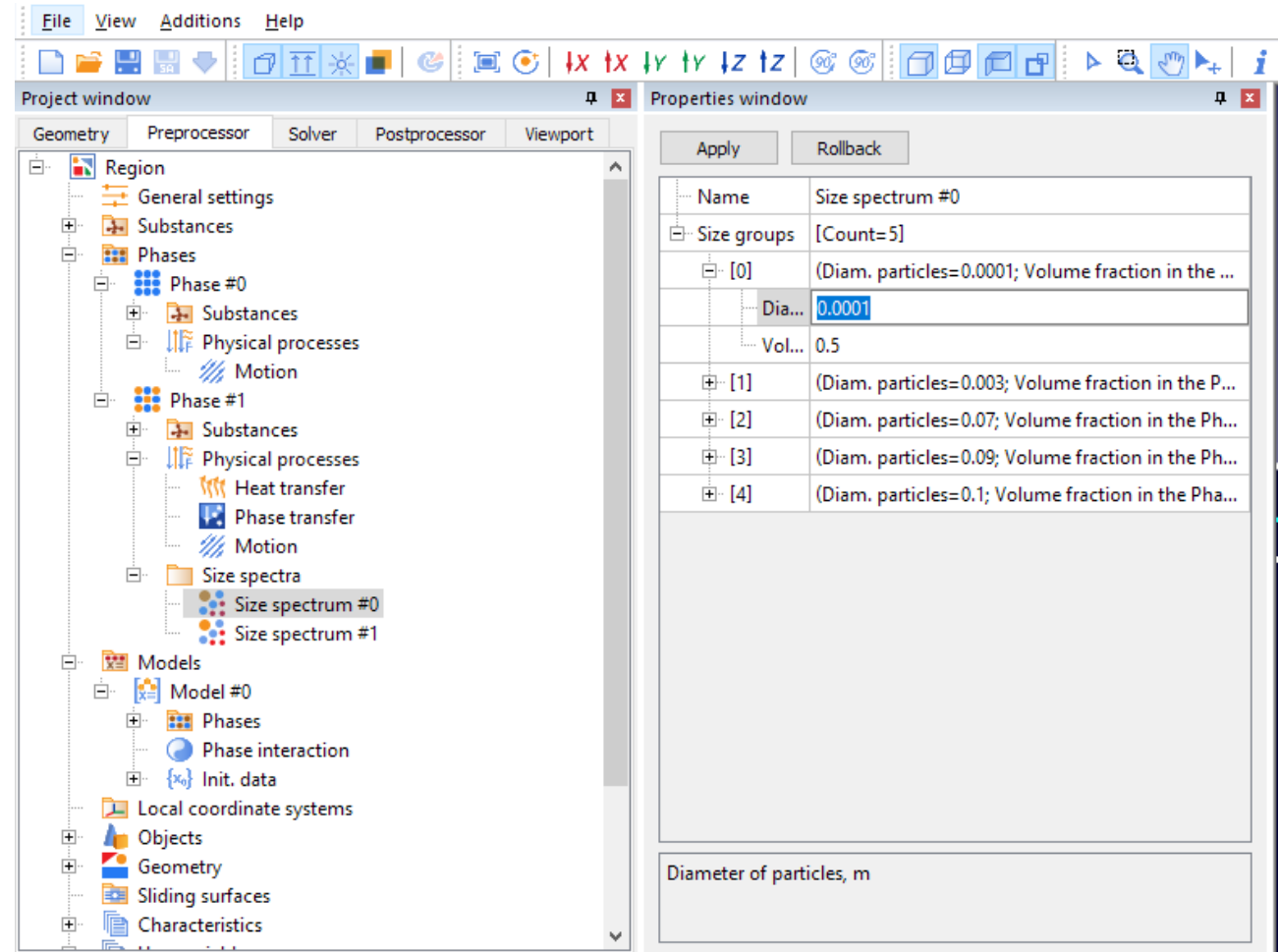
Валидация модели обледенения



NACA 0012, соответствующие условиям:
Run 18-19: $C=0.9144$ м, $V=92.52$ м/с, $OAT=-9.15$ °C, $LWC=0.6$ г/м³, $MVD=80$ мкм, $TIME=13.6$ мин

Дисперсная фаза теперь имеет спектр

- В дисперсной фазе вы теперь можете задать ее спектр по диаметру и объемной концентрации.
- Количество размерных семейств ограничено только вашими вычислительными ресурсами!



The screenshot displays the FlowVision software interface. The 'Project window' on the left shows a tree view of the simulation setup. Under 'Region', 'Phase #0' is expanded to show 'Substances', 'Physical processes' (Motion), and 'Size spectra'. 'Size spectrum #0' is selected. The 'Properties window' on the right shows the configuration for 'Size spectrum #0'. It includes a table of size groups and a text field for the particle diameter.

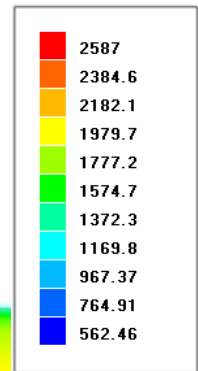
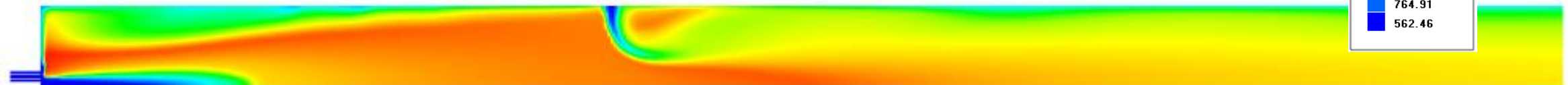
Name	Size spectrum #0
Size groups	[Count=5]
[0]	(Diam. particles=0.0001; Volume fraction in the ...)
Dia...	0.0001
Vol...	0.5
[1]	(Diam. particles=0.003; Volume fraction in the P...)
[2]	(Diam. particles=0.07; Volume fraction in the Ph...)
[3]	(Diam. particles=0.09; Volume fraction in the Ph...)
[4]	(Diam. particles=0.1; Volume fraction in the Pha...)

Diameter of particles, m

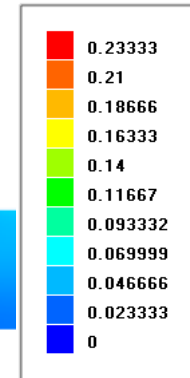
Модель горения угля

- Дисперсная фаза
- Спектр дисперсной фазы по диаметру
- Три фазы горения:
 - Выход воды (сушка)
 - Пиролиз (выход летучих газов и их горения)
 - Горение коксового остатка
- Радиационная модель – либо P1 либо спектральная MDO

Температура

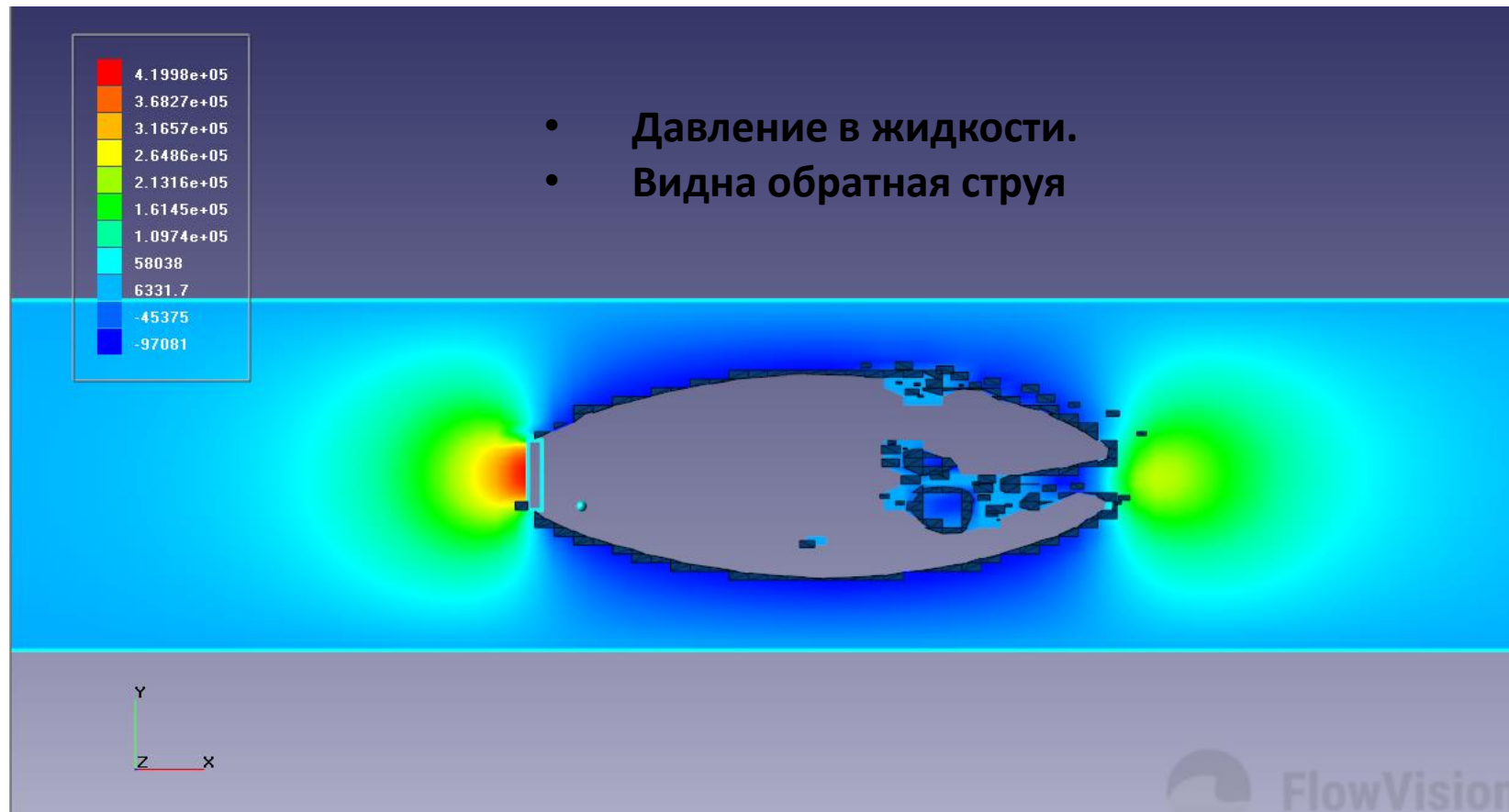


Массовая доля кислорода:



Бета-версия модели кавитации

- В режиме работы VOF «сплошная-вакуум» можно задать модель кавитации.
- Пока работаем в бета-режиме.



Итак, основные новшества:

Модель падения потенциала на границе раздела плазма-металл

Очень вязкие течения

Спектральная модель МДО для теплообмена излучением

Спектральная модель движения дисперсных частиц

Спектральная модель горения угля

Модель обледенения

Модель кавитации (бета)

Вставка начальных условий и модификаторов в режиме с обрезкой

И, как обещал - что внутри FV

FlowVision написан на с++

Главная идея – масштабируемость разработки:

- мало разработчиков
- Мало ресурсов (==денег, мало денег, мало программистов)
- Нет возможности содержать программистов и физиков-математиков, поэтому у нас программист зачастую – универсал
- Демократия куда идти, авторитария когда идти
- Главное – идеи и люди. Пока мы на стадии, когда создатели программы определяют ее успех. (2-ая стадия СМК)

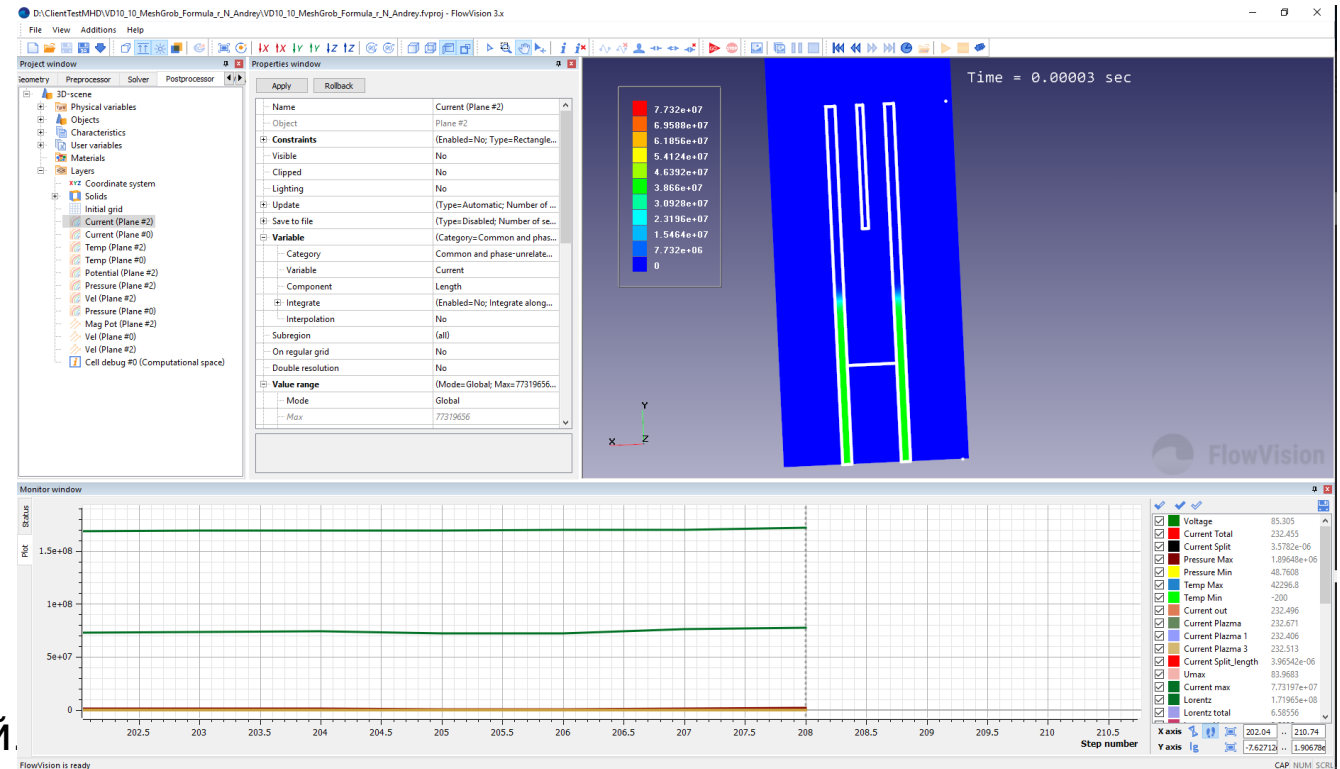
Архитектура FV

Клиент-Серверная

- FVPPP – пре-постпроцессор
- Загрузка геометрической модели
- Составление матмодели
 - Выбор уравнений
 - Расстановка ГУ
 - Начальные условия
- Постпроцессинг (рендеринг)
- Fvsolver – только решение уравнений.

Запускается удаленно.

- Генерация расчетной сетки
- Решение уравнений
- Подготовка данных для постпроцессинга



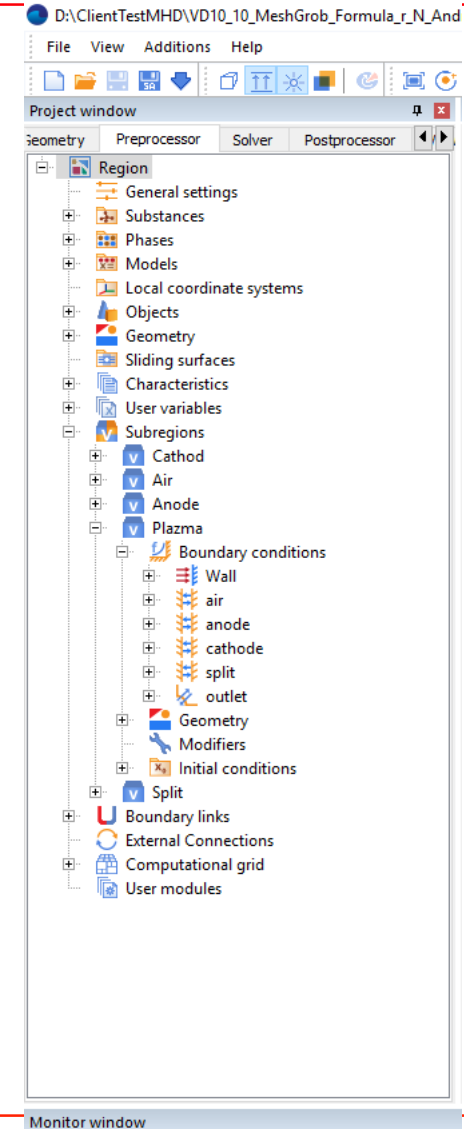
```

C:\Program Files\Microsoft MPI\Bin\mpiexec.exe
*SEND* for CIntegralCharsLayer, GUID=E6E60CA4-61BE-3376-BEAE-041178F9CB19
[layer data built]
*SEND* for CIntegralCharsLayer, GUID=3F9089E1-1608-C587-5E1C-C4E6725C9DE
[layer data built]
*SEND* for CIntegralCharsLayer, GUID=9D194109-D9A3-EEA7-43DA-9A7AB67A83A6
[layer data built]
*SEND* for CIntegralCharsLayer, GUID=99264EAD-81AD-713F-F4CE-4C0689D02EEF
[layer data built]
*SEND* for CIntegralCharsLayer, GUID=990E4CAA-BA70-064D-96A9-77A466AD7414
[layer data built]
*SEND* for CIntegralCharsLayer, GUID=AD85ED92-43D1-6D2F-E8E2-736DE0E78D35
[layer data built]
End: ===== Processing Layers =====; Timer: cpu=7.266 wall=3.276
    
```

Архитектура классов FV повторяет во многом интерфейс

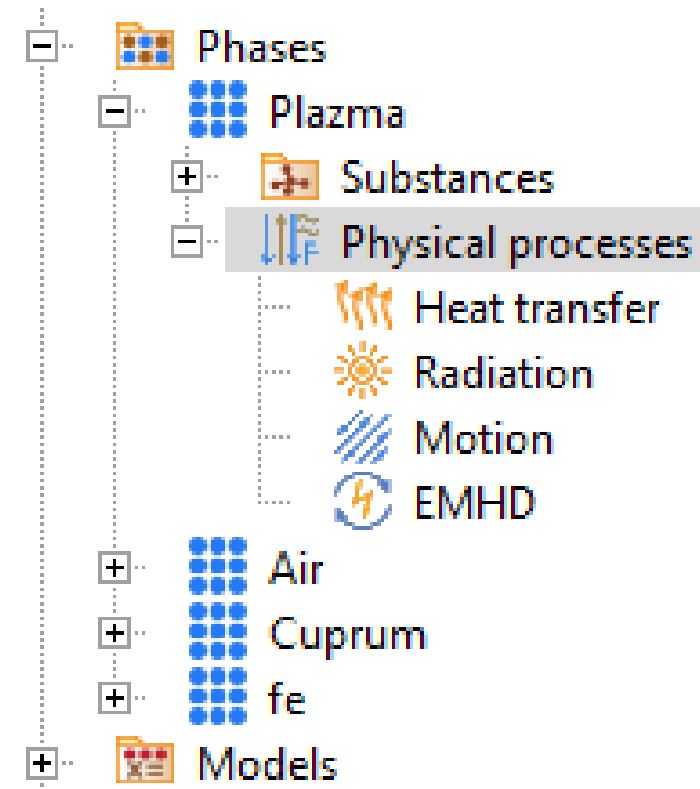
Класс CRegion - основной

- класс CSubstance
 - свойства веществ
- класс CPhase
 - Набор веществ
 - Признак Континуальная/дисперсная
 - Набор уравнений
- Модель
 - Набор фаз
 - Интерфаза == определяет физику взаимодействия фаз



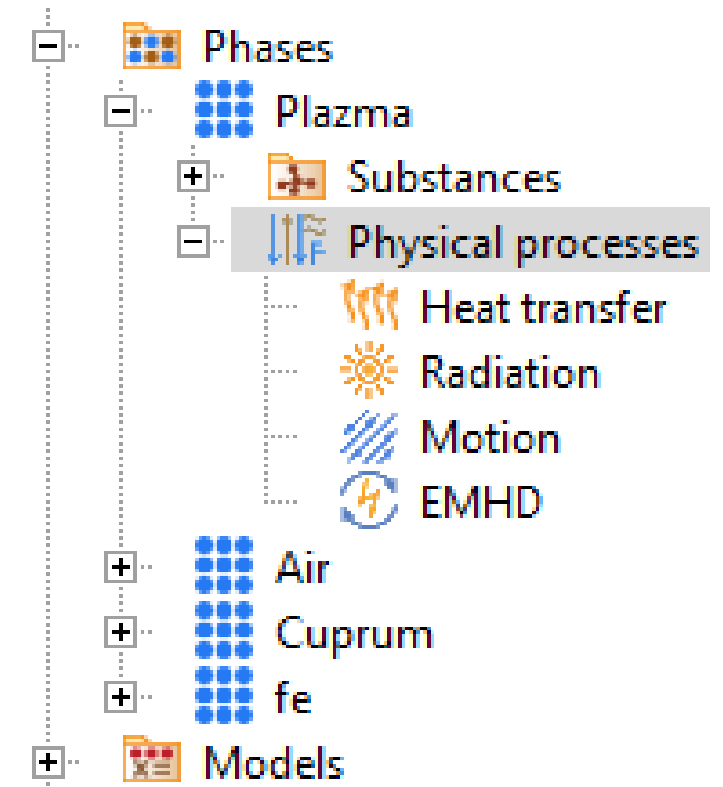
Архитектура классов FV повторяет во многом интерфейс

- класс CPhase
 - Набор веществ
 - Признак Континуальная/дисперсная
 - Набор уравнений
- Решаемую систему уравнений определяет «блок»
Во FlowVision Блок – это класс, определяющий какую-то систему уравнений
 - Основные базовые классы
 - CMotion
 - CHeatTransfer
 - CRadiation
 - ...



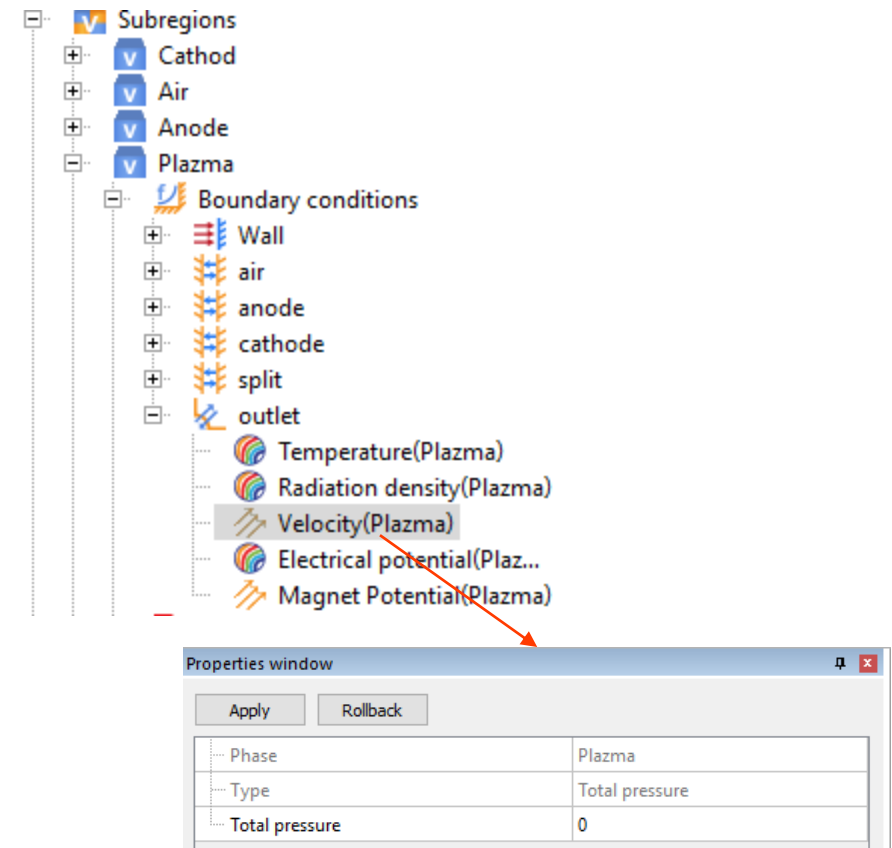
Пример блока, описываемого классом CMotion

- Является базовым для классов, реализующих моделирование:
 - Уравнений Навье-Стокса
 - Уравнений Дарси
- Блок не содержит переменные, он только их знает. Все переменные имеют имена например – VEL – скорость, PRES – давление.
- Сами переменные описаны базовым классом CFVVariable, он содержит реальное имя переменной («Давление», «Скорость») и номер своего поля в ячейке
- Переменные бывают реальные (запоминаемые), имеющие различные флаги – записывать на диск, имеют ГУ, показываются в PPP
 - Скорость – реальная переменная, тип вектор, имеет явное ГУ (есть в интерфейсе), показывается в FVPPP
 - Давление - реальная переменная, тип скаляр, имеет ГУ (скрытое в интерфейсе), показывается в FVPPP
- Есть также виртуальные переменные – они вычисляются «на лету», могут показываться в интерфейсе. Пример – число Маха (скаляр), Q-критерий (вектор)



Пример блока, описываемого классом CMotion

- Блок содержит много методов (==подпрограмм для любителей фортрана)
- Главный метод Блока – это функция
 - ::ScalarSourceTermBasic – правая часть уравнения для давления
 - ::VectorSourceTermBasic – правая часть уравнения Навье-Стокса
- С классами, описывающими уравнения, ассоциирована структура классов
 - CBVar – классы граничных условий. Но по сути, эти классы задают потоки между расчетными ячейками. В этих классах есть два главных метода
 - ConvectiveFlow – конвективный перенос с грани ячейки
 - DiffusionFlux – диффузионный перенос на грани ячейки
 - ScalarBoundaryValue/ VectorBoundaryValue – значение переменной на грани ячейки
 - Если ГУ есть видимое ГУ в интерфейсе, то для него определены имена переменных



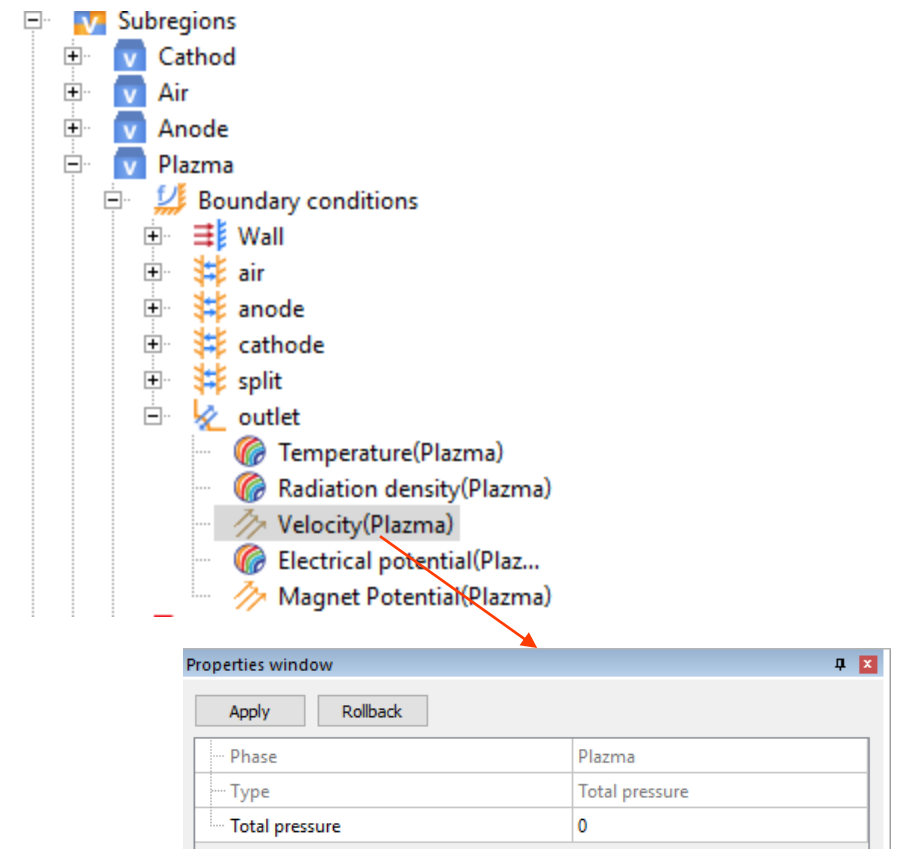
The screenshot shows a tree view of the simulation setup. Under 'Subregions', there are 'Cathod', 'Air', 'Anode', and 'Plazma'. Under 'Boundary conditions', there are 'Wall', 'air', 'anode', 'cathode', 'split', and 'outlet'. The 'outlet' boundary condition is expanded to show 'Temperature(Plazma)', 'Radiation density(Plazma)', 'Velocity(Plazma)', 'Electrical potential(Plazma)', and 'Magnet Potential(Plazma)'. A red arrow points from 'Velocity(Plazma)' to the 'Properties window' below.

The 'Properties window' for 'Velocity(Plazma)' shows the following settings:

Properties window	
Phase	Plazma
Type	Total pressure
Total pressure	0

Пример блока, описываемого классом CMotion

- Блок содержит много методов (==подпрограмм для любителей фортрана)
- Главный метод Блока – это функция
 - ::ScalarSourceTermBasic – правая часть уравнения для давления
 - ::VectorSourceTermBasic – правая часть уравнения Навье-Стокса
- С классами, описывающими уравнения, ассоциирована структура классов CBVar – классы граничных условий. Но по сути, эти классы задают потоки между расчетными ячейками. В этих классах есть два главных метода
 - ConvectiveFlow – конвективный перенос с грани ячейки
 - DiffusionFlux – диффузионный перенос на грани ячейки
 - ScalarBoundaryValue/ VectorBoundaryValue – значение переменной на грани ячейки
 - Если ГУ есть видимое ГУ в интерфейсе, то для него определены имена переменных



The screenshot shows a tree view of the simulation setup. Under 'Subregions', there are 'Cathod', 'Air', 'Anode', and 'Plazma'. Under 'Boundary conditions', there are 'Wall', 'air', 'anode', 'cathode', 'split', and 'outlet'. The 'outlet' boundary condition is expanded to show 'Temperature(Plazma)', 'Radiation density(Plazma)', 'Velocity(Plazma)', 'Electrical potential(Plazma)', and 'Magnet Potential(Plazma)'. A red arrow points from 'Velocity(Plazma)' to the 'Properties window' below.

The 'Properties window' for 'Velocity(Plazma)' shows the following settings:

Properties window	
Phase	Plazma
Type	Total pressure
Total pressure	0

Что дает архитектура классов

- Если уравнение есть уравнение типа конвективно-диффузионного переноса (любое слово можно вычеркнуть), создание новой модели занимает для опытного пользователя неделю
- Программист-физик не заморачивается по поводу интерфейса. Он просто описывает интерфейс макросами типа и значения констант «коэффициент теплообмена» «степень черноты» и «температура стенки» тут же появятся в FVPPP и будут записываться на диск

```
FV_BEGIN_CLASS(CBVarHeatEmission)
    STRUCTURE<SFvValue>(_T("Alpha"), &CBVarHeatEmission::alpha, IDSN_HEATTRANSFERCOEF, IDSD_HEATTRANSFERCOEF, PF_DEFAULT);
    STRUCTURE<SFvValue>(_T("Blackness"), &CBVarHeatEmission::blackness, IDSN_BLACKNESS, IDSD_BLACKNESS, PF_DEFAULT);
    STRUCTURE<SFvValue>(_T("WallTemp"), &CBVarHeatEmission::temp_inf, IDSN_WALLTEMP, IDSD_WALLTEMP, PF_DEFAULT);
FV_END_CLASS
```

! Структура `SFvValue` говорит, что переменная может быть константой, уравнением, таблицей или подключаемой DLL

Хранение данных

- Если фаза во FlowVision – это набор уравнений, то модель – это набор фаз
- Функции модели – управлять данными.
- Все данные (поля скорости, давления, температуры и т.д.) хранятся в хранилище ГиперЯчейка в виде массивов. Модель знает, где для какой ячейки начинаются ее данные и сколько их. Доступ к данным организован так

```
CFVVariable* pvel = model->GetVariable(pEqSys, VEL); // pEqSys - указатель на Блок  
CVector3      vel  = pvel->GetVector(pCell);         // pCell  - указатель на ячейку
```

- Также в ГиперЯчейке хранятся объекты, описывающие стороны ячейки (Chip-ы в терминологии FlowVision) + данные на самих Chip.
- Гиперячейка – это основной структурный элемент расчетной сетке, по которому происходит декомпозиция расчетной области между процессорами.
- Пробег по нитям происходит независимо для каждой гиперячейки

Решение уравнений

- Для того, чтобы провести какую-то операцию с ячейками, нужно составить инструкцию для алгебраического решателя
- Записываем инструкцию

```
case CEquationType::SOLVE_ELECTRO_EQ:  
{  
    // Вычисление проводимости  
    eqType = new CEquationType();  
    eqType->block    = this;  
    eqType->action   = CONDUCTIVITY_CALC;  
    pvar[0] = _model->GetVariable(this, CONDUCTIVITY);  
    eqType->SetVariable(pvar, 1);  
    buf.Insert(eqType);  
}
```

- Пишем код выполнения

```
case CONDUCTIVITY_CALC:  
{  
    CFVVariable* cond_var = _eqType->GetVariable(_pCell, 0);  
    conductivity = phase->GetSubstances()[0]->GetValue(CSubstance::COND, presAbs, tempAbs, _pCell, phase);  
    cond_var->SetScalar(_pCell, conductivity);  
    return;  
}
```

Решение уравнений

- Для того, чтобы решить уравнение, нужно составить инструкцию, например
- Записываем инструкцию

```
case CEquationType::SOLVE_ELECTRO_EQ:
{
    //Решение уравнения для потенциала
    eqType = new CEquationType();
    pvar[0] = _model->GetVariable(this, POT);
    eqType->SetVariable(pvar, 1);
    eqType->fluxType = CEquationType::DIFFUSIVE;
    CMethod* meth = new CMethod(this, _model, DENS, POT_CURR);
    eqType->SetMethod(meth, 0);
    eqType->residual = _model->GetVariable(this, POT_CURR);
    eqType->initialSolverThreshold = 3;
    buf.Insert(eqType);
}
```

Все инструкции имеют имена и вызываются в некотором порядке, который задает программист

