Лекция 2.2. Методы решения систем уравнений. Критерий итерационной сходимости.

- Метод конечных разностей;
- **Метод контрольных объемов**;
- Метод конечных элементов;
- Метод сглаженных частиц;
- Метод с использованием функции распределения вероятностей.

Метод контрольных объемов

Дискретизация *– преобразование непрерывной функции в дискретную.*

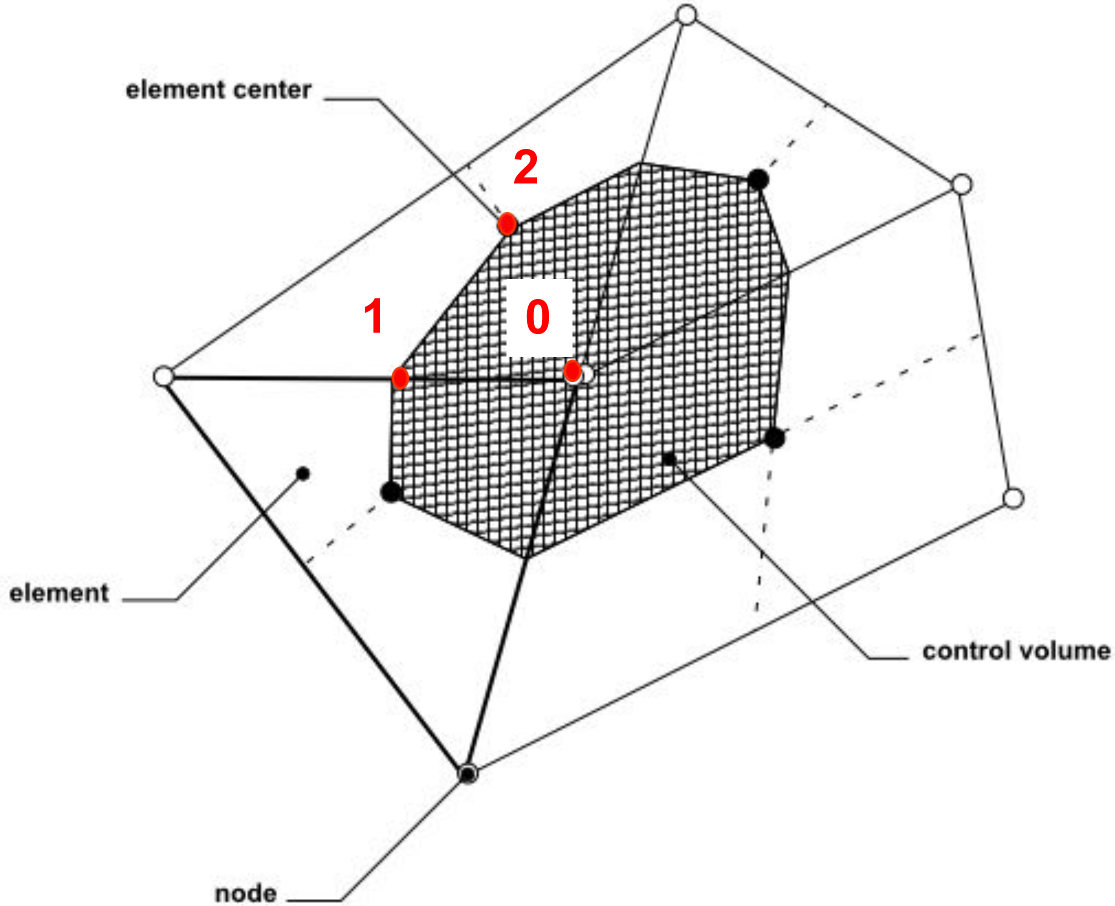
ANSYS CFX использует метод конечных объемов на основе элементов дискретизации пространственной области с использованием сетки. Сетка нужна для построения конечных объемов, которые используются для применения законов сохранения соответствующих величин, таких как масса, импульс и энергия. Сетка трехмерна, но для простоты рассмотрим двухмерную.

Построение сеточной модели – дискретизация пространства.

Задание временного шага – дискретизация времени.

Типичная двумерная сетка

Все переменные решения и свойства текучей среды хранятся в узлах ***Node*** (вершины сетки). Контрольный объем ***Control Volume*** (заштрихованная область) строится вокруг каждого узла сетки следующим образом: контрольный объем ограничивается линиями, соединяющими центры ребер (т. 1) и центры граней ***Element Center*** (т. 2) сеточных элементов *Element*, окружающих уз



Методология метода конечного объёма

Для иллюстрации методологии метода конечного объема рассмотрим уравнения сохранения массы, импульса, выраженные в декартовых координатах:

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho v_i)}{\partial x_i} = 0,$$

$$\frac{\partial (\rho v_{\boldsymbol{i}})}{\partial t} + \frac{\partial (\rho v_i v_j)}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\mu_{eff}\left(\frac{\partial v_{\boldsymbol{i}}}{\partial x_j} + \frac{\partial v_{\boldsymbol{j}}}{\partial x_i}\right)\right)$$

Эти уравнения интегрируются по каждому контрольному объему с использованием теоремы Гаусса о преобразовании объемных интегралов в поверхностные интегралы.

Методология метода конечного объёма

$$\frac{\partial}{dt}\int_V \rho dV + \int_S \rho v_i\, dn_i = 0,$$

$$\frac{\partial}{dt}\int_V \rho v_i dV + \int_S \rho\, v_j v_i dn_i = -\int_S P\, dn_i + \int_S \mu_{eff}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) dn_i + \int_V S v_i dV$$

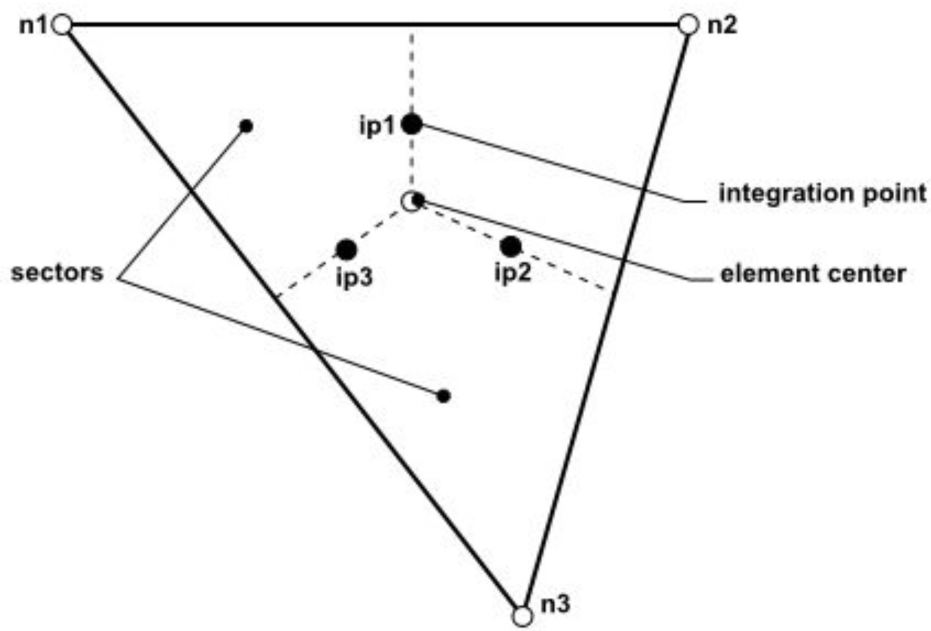
где V и S соответственно, объемные и поверхностные области интегрирования, а dn_i - дифференциальные декартовы компоненты внешнего нормального поверхностного вектора.

Следующим шагом в численном алгоритме является дискретизация объемных и поверхностных интегралов.

Методология метода конечного объёма

Объемные интегралы дискретизируются в каждом секторе ***Sector*** сеточного элемента ***Element*** и накапливаются в контрольном объеме ***Control Volume***, к которому принадлежит сектор.

Поверхностные интегралы дискретизируются в точках интегрирования ($\boldsymbol{ip_n}$), расположенных в центре грани каждого сегмента сеточного элемента.



Методология метода конечного объёма

После дискретизации объемных и поверхностных интегралов интегральные уравнения преобразуются:

$$V\left(\frac{\rho-\rho^0}{\Delta t}\right)+\sum\nolimits_{ip}\dot{m}_{ip}=0,$$

$$V\left(\frac{\rho v_i-\rho^0 v^0{}_i}{\Delta t}\right)+\sum_{ip}\dot{m}_{ip}(v_i)_{ip}=\sum_{ip}(P\Delta n_i)_{ip}+\sum_{ip}\left(\mu_{eff}\left(\frac{\partial v_i}{\partial x_j}+\frac{\partial v_j}{\partial x_i}\right)\Delta n_j\right)_{ip}+\overline{S_\varphi}V$$

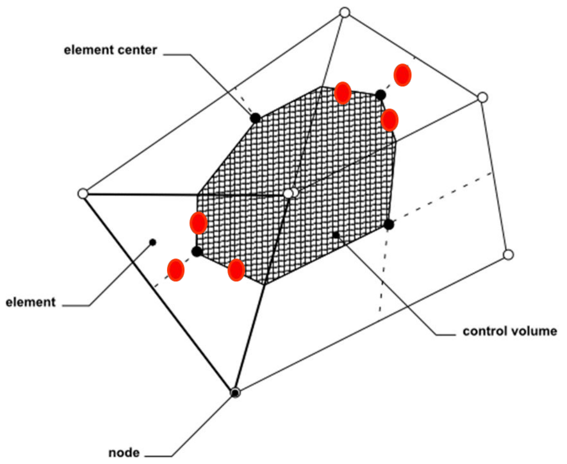
где V – контрольный объем;

Δt – шаг по времени;

Δn_i – дискретный нормальный вектор к внешней поверхности;

индекс ip обозначает вычисления в точке интегрирования, суммированные по всем точкам интегрирования контрольного объема;

верхний индекс 0 – указывает на предыдущий итерационный шаг.



Решение линеаризованных уравнений (метод итерационного приближения)

Итерационные алгоритмы

Итерационными (пошаговыми) алгоритмами *называются алгоритмы, в которых на каждом шаге используется одна и та же формула, выраженная через значения, полученные на предыдущих шагах алгоритма.*

Выполнение такого алгоритма сводится к генерации некоторой числовой последовательности результатов:

$$(x_0, x_1, x_2, \ldots, x_к, x_{к+1}, \ldots),$$

где ***к*** – номер итерации, $x_к$ - значение, полученное на ***к***-м шаге процесса.

Итерационная последовательность своим пределом должна иметь искомое значение ***x****:

$$\lim_{к \to \infty} x_к = x^*.$$

Если такой предел существует, то итерационный процесс называется ***сходящимся****. Если нет, то* ***расходящимся****.*

Критерий итерационной сходимости

Реальный вычислительный процесс всегда должен заканчиваться при конечном значении **k**, поэтому возникает проблема выбора условия окончания итераций – величины **критерия сходимости Δ**.

1. ***Абсолютное*** изменение параметра на соседних шагах итерационного процесса

$$| x_k - x_{k-1} | \le \Delta;$$

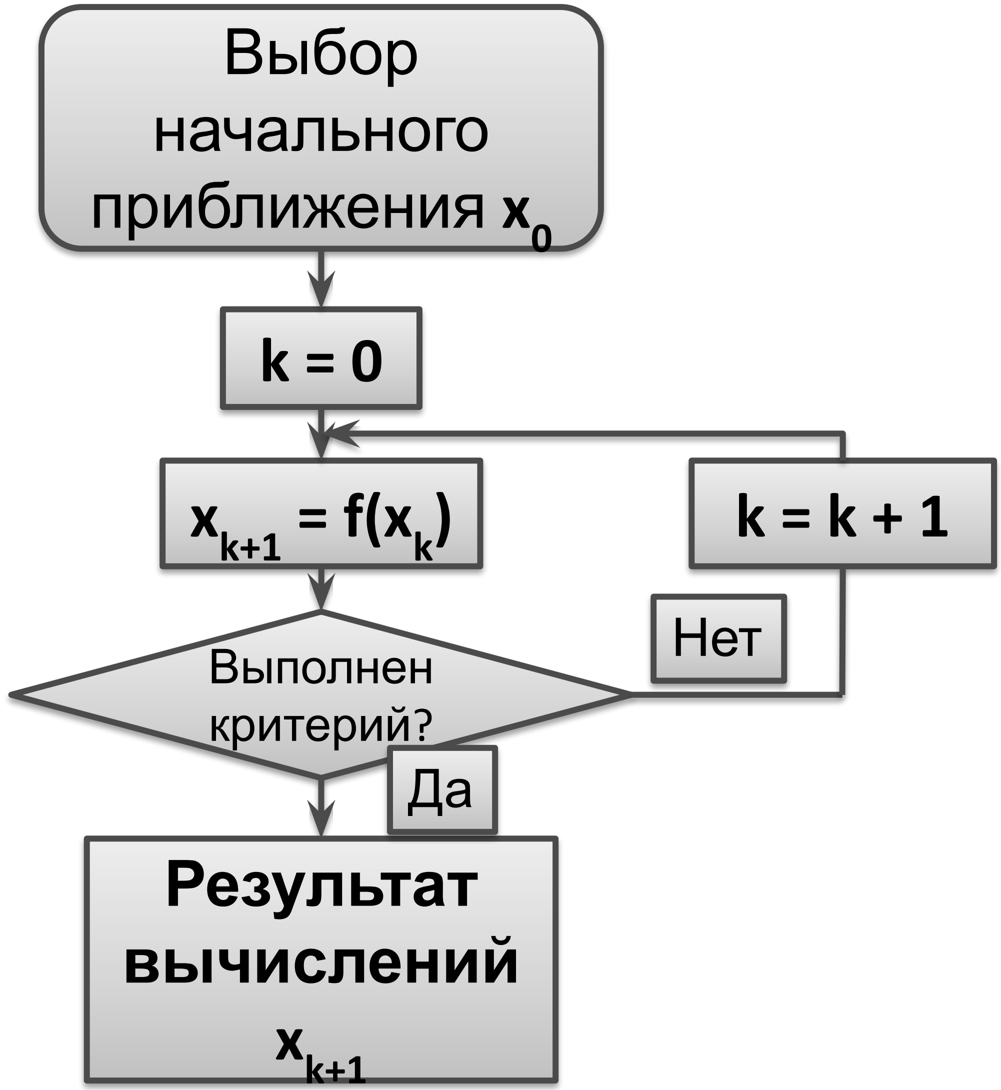
2. ***Относительное*** изменение параметра на соседних шагах

$$| (x_k - x_{k-1}) / x_k | \le \Delta;$$

где **Δ** – заданное пользователем малое значение, определяющая точность нахождения решения.

Критерий итерационной сходимости *– мера локального дисбаланса или невязка каждого уравнения в контрольном объеме.*

Общая блок-схема итерационных алгоритмов



Выбор величины критерия итерационной сходимости

Численное решение уравнений до достижения установленного критерия итерационной сходимости ***Δ*** определяет точность расчета:

Δ > 10^{-4} – достаточная точность для получения качественного понимания поля течения;

Δ = 10^{-4} – относительно неточный расчет, но может быть достаточным для многих инженерных задач. Эта величина по умолчанию установлена в ANSYS CFX.

10^{-4} < ***Δ*** < 10^{-6} – хорошая сходимость, и, как правило, достаточная для большинства технических задач.

Δ ≤ 10^{-6} – точный расчет, применяется для геометрически чувствительных элементов (расчета в переходных областях при резком сужении или расширении канала, при расчете пограничного слоя и т.д.). Зачастую на практике невозможно достичь такого уровня точности.

Реализация итерационного алгоритма в ANSYS CFX

Решение набора линеаризованных уравнений для каждого контрольного объема на каждом итерационном шаге:

$$[\mathbf{A}][\boldsymbol{\varphi}]=[\mathbf{b}],$$

где [A] – коэффициенты перед неизвестными;

[φ] – неизвестные;

[b] – свободные члены.

Пусть ϕ_0 – начальное приближение для неизвестных;

ϕ' – поправка решения;

n – текущий шаг интегрирования.

Система может быть решена итеративно с использованием начального приближения, которое корректируется поправкой на каждом шаге для достижения более точного значения:

$$\varphi^{n+1} = \varphi^{n} + \varphi',$$

где φ’ – решение следующего уравнения,

$$A\varphi' = b - A\varphi^{n}.$$

При повторении указанных действий решение достигает требуемого уровня точности Δ, определённого пользователем:

$$\Delta = \frac{\sum_{i=1}^{N}\left|\phi_i^{n+1} - \phi_i^{n}\right|}{\sum_{i=1}^{N}\left|\phi_i^{n+1}\right|},$$

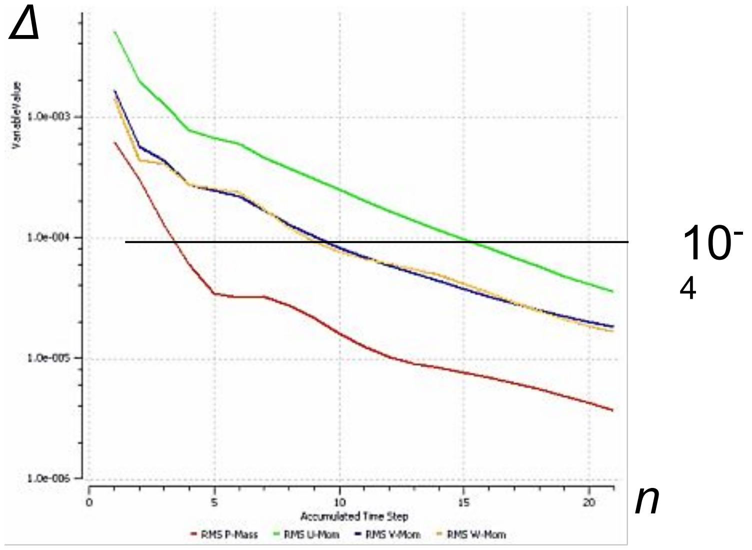
где n – номер итерации;

N – общее число конечных элементов;

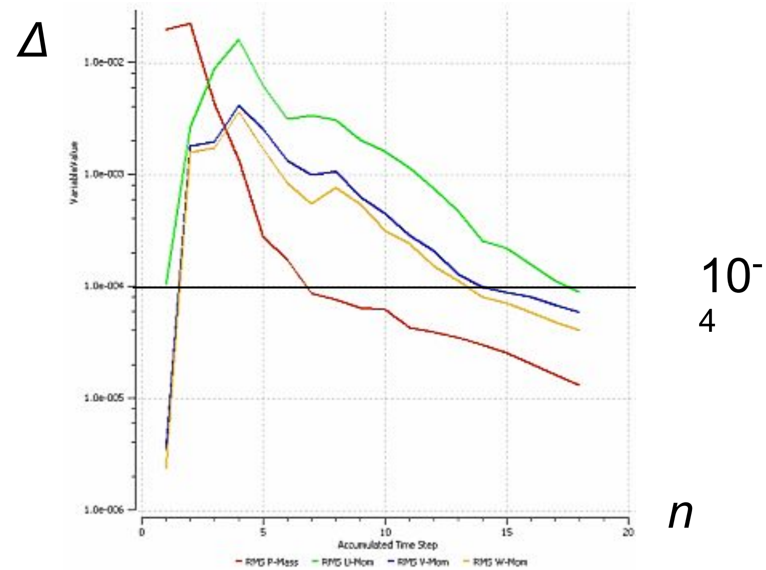
φ – решение.

Графики итерационной сходимости

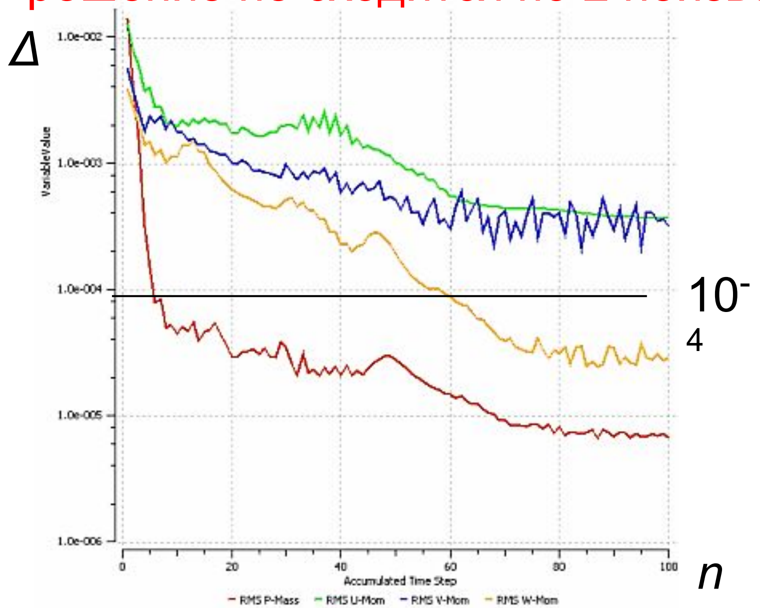
1 – идеальная сходимость



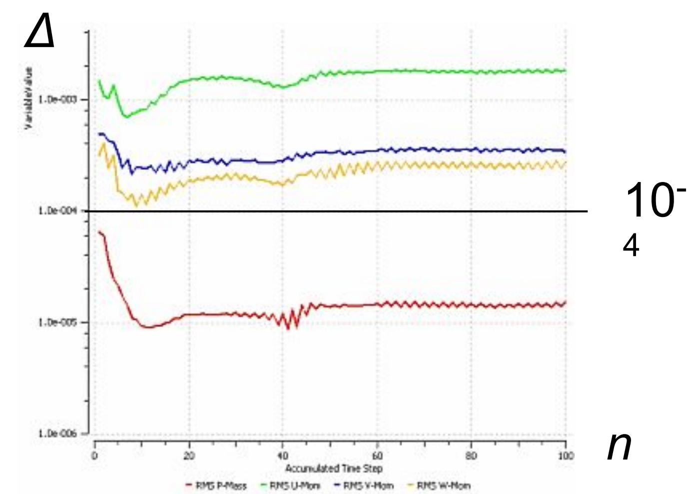
2 – хорошая сходимость



3 – решение не сходится по 2 неизвестным



4 – сходимость отсутствует



Устранение проблем со сходимостью

Если имеются проблемы со сходимостью, необходимо найти их источник, не принимая полученные результаты.

В первую очередь надо понять какой характер она носит ошибка, глобальный или локальный.

1. Сравните RMS (средние) и MAX (максимальные) невязки уравнений, имеющих плохую сходимость.

Если MAX невязка превышает RMS более чем на порядок, это обычно свидетельствует о локальной проблеме сходимости (сетка, ГУ, НУ).

2. Выяснение расположения этой локализации в расчетной области является первым этапом решения проблемы. Для этого в постпроцессоре необходимо создать локализацию (например, изоповерхность) с невязкой (Resedual) в качестве переменной. Чтобы получить массив невязок в файле результатов необходимо в постпроцессоре в объекте Output Control задействовать соответствующую опцию (Results/Output Equation Reseduals/All).

3. Если область с максимальными невязками находится далеко как от интересующей области, так и от выходной границы (Outlet), то решение можно считать корректным.

4. При глобальной проблеме – необходима корректировка задачи.