

Метод конечных элементов в CFD.

Часть 1

Наумкин Виктор Сергеевич

Лекция №10, 2021

Дискретный аналог уравнений Навье-Стокса

Расчёт объёма ячейки

Green-Gauss Cell-Based and Node-Based Gradient Schemes
Green-Gauss Method

Least-Squares Gradient Scheme

Дискретный аналог уравнений Навье-Стокса

Уравнения Навье-Стокса

$$\frac{\partial (\rho U)}{\partial t} + \nabla \cdot (\rho U U) = -\nabla p + \nabla \cdot (\mu (\nabla U) + (\nabla U)^T) - \frac{2}{3} \mu (\nabla \cdot U) \cdot I + \rho g \quad (1)$$

- ▶ Сформируем матрицу, которую затем будем решать алгоритмом SIMPLE (об алгоритме в следующих лекциях):

$$M \cdot U = B \quad (2)$$

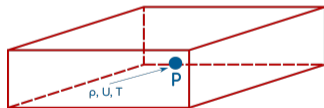
- ▶ M - матрица коэффициентов.

Уравнения Навье-Стокса

Стационарное ламинарное несжимаемое течение

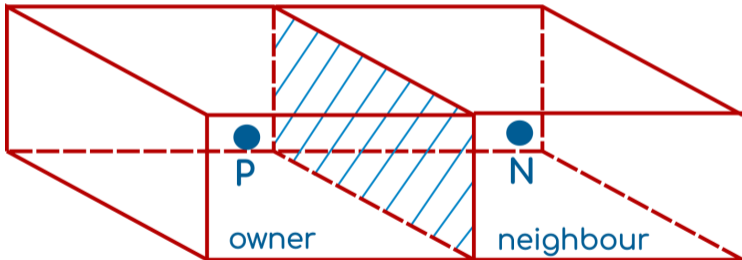
$$\nabla \cdot (UU) = -\frac{1}{\rho} \nabla p + \nabla \cdot (\nu \nabla U) + g \quad (3)$$

- ▶ Проведём дискретизацию на 3D полиэдральной сетке.
- ▶ В МКЭ второго порядка переменные потока изменяются линейно по ячейке.



- ▶ Значения переменных рассчитываются в центре ячейки (P).

- ▶ Рассмотрим соседние ячейки.
- ▶ Значения переменных хранятся в центрах данных ячеек (P, N).
- ▶ В общем виде имеется M - соседних ячеек.



Интегрирование

- ▶ Проинтегрируем уравнения Навье-Стокса по ячейке P :

$$\int_V \left[\nabla \cdot (UU) + \frac{1}{\rho} \nabla p - \nabla \cdot (\nu \nabla U) - g \right] dV = 0 \quad (4)$$

- ▶ Проинтегрируем каждый член по отдельности:

$$\int_V [\nabla \cdot (UU)] dV = \int_V \left[-\frac{1}{\rho} \nabla p \right] dV + \int_V [\nabla \cdot (\nu \nabla U)] dV + \int_V [g] dV = 0 \quad (5)$$

Постоянные источниковые члены

- ▶ Для силы тяжести:

$$\int_V [g] dV = gV_P \quad (6)$$

- ▶ V_P - объём ячейки P .
- ▶ \Rightarrow в матричной форме:

$$M \cdot U = B \quad (7)$$

- ▶ Гравитационная сила добавляется в правую часть (в вектор B).

Линейные источникные члены

- ▶ Предположим, что источникный член SU добавлен в уравнения Навье-Стокса. (S - некоторый скаляр)

$$\nabla \cdot (UU) = -\frac{1}{\rho} \nabla p + \nabla \cdot (\nu \nabla U) + g + \underbrace{SU}_{\text{source term}} \quad (8)$$

Линейные источниковые члены

$$\int_V [SU] dV = S_P \int_V U dV = \quad (9)$$

$$= S_P \int_V (U_P + (x - x_P) (\nabla U_P)) dV = \quad (10)$$

$$= S_P U_P \int_V dV + \underbrace{\left[\int_V (x - x_P) dV \right]}_{=0} \cdot \nabla U_P = \quad (11)$$

$$= S_P U_P V_P \quad (12)$$

- ▶ Можно добавить $S_P V_P$ в матрицу M .
- ▶ или добавить $S_P V_P V_P$ в вектор-столбец B .

Явный/неявный методы

- ▶ добавление в матрицу M - это неявный метод.
- ▶ добавление в вектор-столбец B - это явный метод.

$$M = \begin{vmatrix} -S_1 V_1 & 0 & \dots & 0 \\ 0 & -S_2 V_2 & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & -S_M V_M \end{vmatrix} \quad (13)$$

$$B = \begin{vmatrix} S_1 U_1 V_1 \\ S_2 U_2 V_2 \\ \vdots \\ S_M V_M \end{vmatrix} \quad (14)$$

Смешанный явный/ неявный методы

- ▶ Если имеется источник в виде SU^2

$$\text{или } (SU^{i-1}) \cdot U \quad (15)$$

- ▶ \Rightarrow интегрирование даст:

$$\int_V SU^2 dV = S_P U_P^{i-1} U_P \quad (16)$$

- ▶ Можно использовать значение U^{i-1} с предыдущей итерации для линеаризации источникового члена.

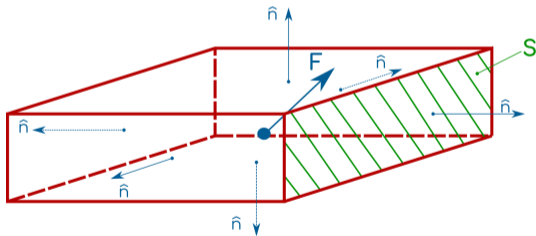
Конвективные и диффузионные члены

- Данные члены содержат оператор дивергенции ($\nabla \cdot$)

$$\underbrace{\int_V [\nabla \cdot (UU)] dV}_{\text{Конвекция}} = \int_V \left[-\frac{1}{\rho} \nabla p \right] dV + \underbrace{\int_V [\nabla \cdot (\nu \nabla U)] dV}_{\text{Диффузия}} + \int_V [g] dV = 0 \quad (17)$$

Теорема Гаусса-Остроградского

- ▶ Преобразуем интеграл по объёму в поверхностные интегралы
- ▶ если " F " - вектор, то
$$\int_V \nabla \cdot F dV = \int_S F \cdot \hat{n} dS$$



Конвективные члены

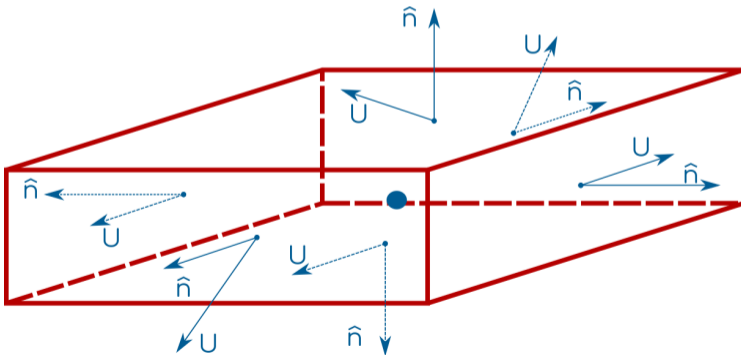
- ▶ Используя теорему Гаусса-Остроградского, конвективный член запишется следующим образом:

$$\int_V [\nabla \cdot (UU)] dV = \int_S [U (U \cdot \hat{n})] dS \quad (18)$$

- ▶ $(U \cdot \hat{n}) dS$ - поток через поверхность S
- ▶ вторая скорость за скобками U - неизвестная скорость, которую необходимо найти.

- Разделим поверхностный интеграл на сумму интегралов через поверхности конечного элемента:

$$\int_S [U (U \cdot \hat{n})] dS = \sum_{i=1}^M \int_S [U_i (U_i \cdot \hat{n}_i)] dS_i \quad (19)$$



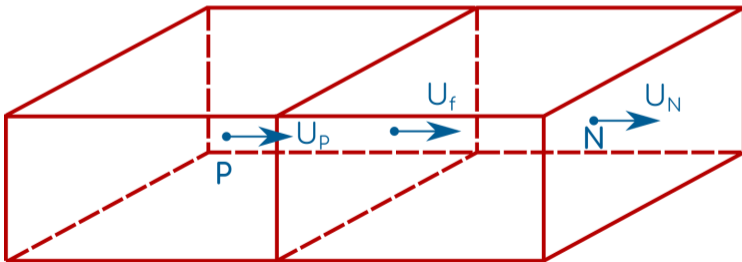
- ▶ Изменение величины по поверхности линейно
- ▶ \Rightarrow значение можно аппроксимировать по значению в центре поверхности f
- ▶ Jasak(1996):

$$\sum_{i=1}^M \int_S [U_i (U_i \cdot \hat{n}_i)] dS_i \approx \sum_{i=1}^M [U_{f_i} (U_{f_i} \cdot n_{f_i})] \cdot S_{f_i} \quad (20)$$

- ▶ Интеграл ушёл.

- ▶ Нужны значения искомой величины в центре поверхности (f), но в CFD рассчитываются и хранятся значения только в центрах ячеек (P, N)
- ▶ \Rightarrow необходимо сделать интерполяцию:

$$\sum_{i=1}^M [U_{f_i} (U_{f_i} \cdot \hat{n}_{f_i})] \cdot S_{f_i} \quad (21)$$



Интерполяция

- ▶ CFD коды обеспечивают следующие схемы интерполяции для расчёта U_f
 - ▶ Upwind
 - ▶ Second order/Linear Upwind
 - ▶ Central Differencing
 - ▶ QUICK
- ▶ Данные схемы рассчитывают значение переменной на поверхности (f) из значений в текущей ячейке (owner) P и соседней с ней (neighbor) N
- ▶ Каждая схема будет рассмотрена дальше по отдельности

Конвективные члены

- ▶ Как упоминалось:

$$\int_V [\nabla \cdot UU] dV \approx \sum_{i=1}^M U_{fi} F_{fi} \quad (22)$$

- ▶ где значение на поверхности ячейки рассчитывается по какой-либо интерполяционной схеме.
- ▶ В матрицу M добавляются данные члены

$$MU = B \quad (23)$$

Заполнение матрицы M

- ▶ Конвективные члены приводят к диагональным и недиагональным элементам матрицы:

$$M = \begin{pmatrix} M_{11} & M_{12} & M_{13} & \cdots & 0 \\ M_{21} & M_{22} & 0 & \cdots & 0 \\ 0 & M_{23} & M_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & M_{mm} \end{pmatrix} \quad (24)$$

Итоги

- ▶ Члены уравнений Навье-Стокса могут быть проинтегрированы независимо друг от друга

$$\nabla \cdot (UU) = -\frac{1}{\rho} \nabla p + \nabla \cdot (\nu \nabla U) + g \quad (25)$$

- ▶ Это приводит к диагональным и недиагональным элементам в матрице M и источникам в матрице B , например:

$$\int_V [\nabla \cdot UU] dV \approx \sum_{i=1}^M U_{f_i} F_{f_i} \quad (26)$$

- ▶ Диффузионные члены?

Расчёт объёма ячейки

Background

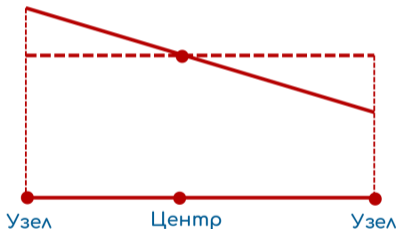
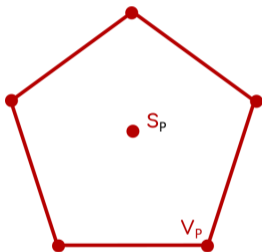
- ▶ Объём ячейки необходим для определения источниковых членов

$$\frac{\partial (\rho k)}{\partial t} + \nabla \cdot (\rho k U) = \nabla \cdot \left(\left(\mu + \frac{\mu_t}{\sigma_k} \right) \nabla k \right) + P_\epsilon - \rho \epsilon + \underline{S_k} \quad (27)$$

- ▶ Интегрирование по контрольному объёму даёт:

$$\int_V [S_k] dV = S_P V_P \quad (28)$$

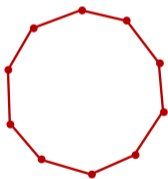
- ▶ Все значения переменных потока изменяются линейно поперёк ячейки \Rightarrow Можно взять значение в центре ячейки (S_P) и умножить на объём V_P
- ▶ Для этого необходимо знать объём ячейки.



- ▶ Интеграл такой же, если взять значение в центре ячейки!

Объём ячейки

- ▶ Необходим метод расчёта, который подходит для:
 1. Ячеек с N сторон (полиэдральные ячейки)
 2. Скошенных ячеек
 3. Ячеек с различным размером сторон



n-сторонняя ячейка



скошенная ячейка



ячейка с различными сторонами

- ▶ Воспользуемся теоремой Гаусса-Остроградского для определения формулы для нахождения объёма ячейки.
- ▶ Данный метод может использоваться для любого N -стороннего полигона.
- ▶ Для любого векторного поля:

$$\int_V [\nabla \cdot B] dV = \int_A [B \cdot \hat{n}] dA \quad (29)$$

- ▶ Скорость аккумуляции в контрольном объёме равна потоку через поверхность.



$$\int_V [\nabla \cdot B] dV = \int_A [B \cdot \hat{n}] dA \quad (30)$$

- ▶ Нужно получить выражение для объёма ячейки V_P .
- ▶ Если принять $\nabla \cdot B = 1$, то

$$\int_V [1] dV = \int_A [B \cdot \hat{n}] dA \quad (31)$$

$$V_P = \int_A [B \cdot \hat{n}] dA \quad (32)$$

▶ Нужно $\nabla \cdot B = 1$.

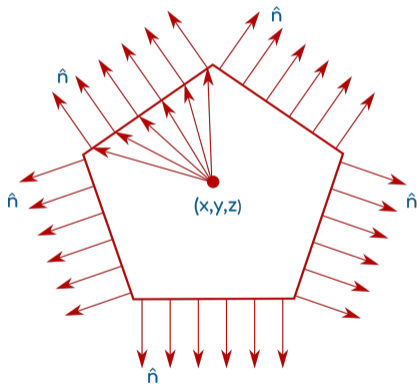
▶ Пусть $B = \frac{1}{3} (x, y, z) \Rightarrow$

$$\nabla \cdot B = \frac{1}{3} (x, y, z) \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right) = \frac{1}{3} (1 + 1 + 1) = 1 \quad (33)$$

$$V_P = \int_A [B \cdot \hat{n}] dA \quad (34)$$

$$V_P = \int_A \left[\frac{1}{3} (x, y, z) \cdot (n_x, n_y, n_z) \right] dA \quad (35)$$

- ▶ Что обозначает интеграл в последнем выражении?
- ▶ Для каждой точки поверхности (x, y, z) проинтегрируем произведение единичного вектора на его местоположение, по всей поверхности:



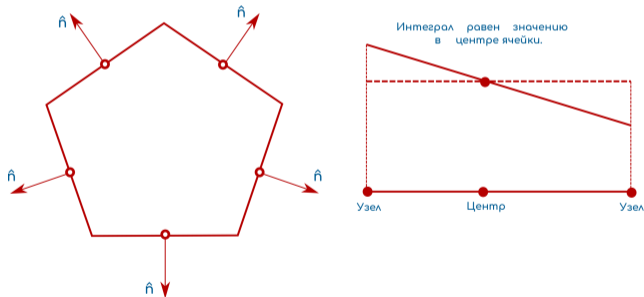
Интегрирование
по всей поверхности

- ▶ Можно расщепить полигон на отдельные стороны и проинтегрировать по отдельности:

$$V_P = \sum_{faces A} \int \left[\frac{1}{3} (x, y, z) \cdot (n_x, n_y, n_z) \right] dA \quad (36)$$

Поверхностный интеграл

- ▶ Все величины изменяются линейно по ячейке и поверхности.
- ▶ ⇒ Можно заменить интеграл по поверхности значением в центре поверхности!



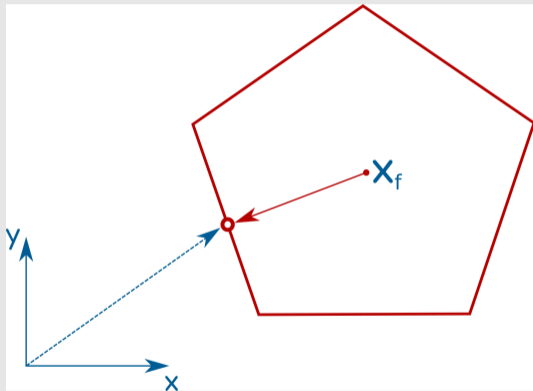
$$V_P = \sum_{faces A} \int \left[\frac{1}{3} (x, y, z) \cdot (n_x, n_y, n_z) \right] dA \quad (37)$$

- Заменяем интеграл на значение в центре поверхности f

$$V_P = \sum_{faces} \left[\frac{1}{3} (x_f, y_f, z_f) \cdot (n_x, n_y, n_z) \right] dA_f \quad (38)$$

$$V_P = \sum_{faces} \left[\frac{1}{3} (x_f \cdot \hat{n}_f) \right] dA_f$$

x_f определяется из центра ячейки. Не из глобальных координат!



для 2D случая

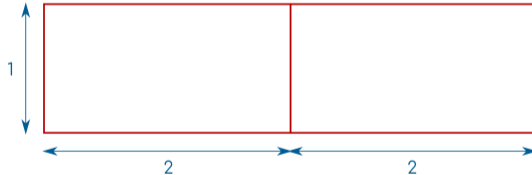
$$B = \frac{1}{2} (x, y) \Rightarrow \nabla \cdot B = 1 \quad (39)$$

$$V_P = \sum_{faces} \left(\frac{1}{2} (x_f \cdot \hat{n}_f) \right) A_f \quad (40)$$

- ▶ Отличие от 3D в коэффициенте $\frac{1}{2}$, в 3D - $\frac{1}{3}$

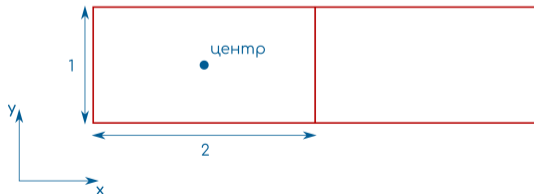
Пример

- ▶ Рассмотрим два прямоугольника и рассчитаем площадь каждого (2)



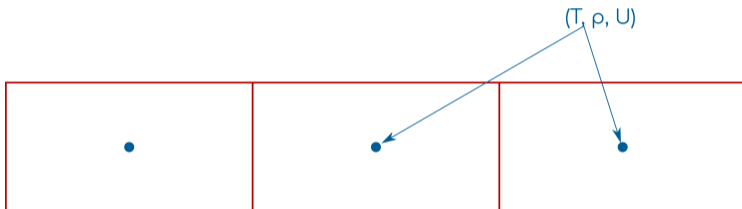
$$V_P = \underbrace{\frac{1}{2} (0; -0.5) (0; -1) \cdot 2}_{\text{bottom}} + \underbrace{\frac{1}{2} (1; 0) (1; 0) \cdot 1}_{\text{right}} + \underbrace{\frac{1}{2} (0; 0.5) (0; 1) \cdot 2}_{\text{top}} + \underbrace{\frac{1}{2} (-1; 1) (0; -1) \cdot 1}_{\text{bottom}} \quad (41)$$

$$V_P = \frac{1}{2} + \frac{1}{2} + \frac{1}{2} + \frac{1}{2} = 2 \quad (42)$$



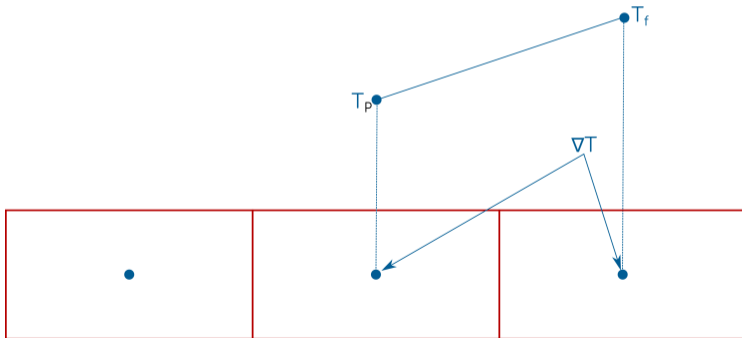
Green-Gauss Cell-Based and Node-Based Gradient Schemes

- ▶ CFD коды хранят значения переменных в центрах ячеек.



- ▶ Необходимы значения градиентов величин в центрах ($\nabla T, \nabla U, \nabla p$).
- ▶ В декартовых координатах:

$$\nabla T = \left(\frac{\partial T}{\partial x}, \frac{\partial T}{\partial y}, \frac{\partial T}{\partial z} \right) \quad (43)$$

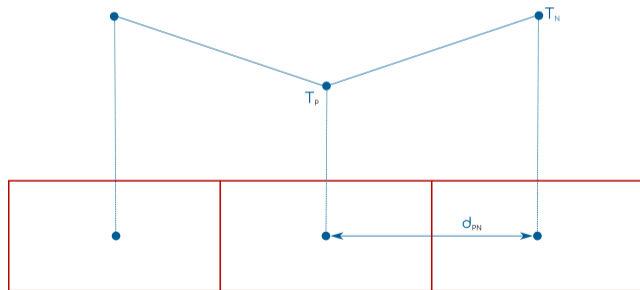


- ▶ Градиенты ($\nabla T, \nabla U, \nabla p$) необходимы для:
 1. Linear Upwind Differencing
 2. Non-orthogonal correctors
 3. Source terms

$$T_f = T_P + (\nabla T)_P \cdot r_{PF} \quad (44)$$

Вычисление градиента: 1D

$$(\nabla T)_f = \left(\frac{T_N - T_P}{d_{NP}}; 0; 0 \right) \quad (45)$$

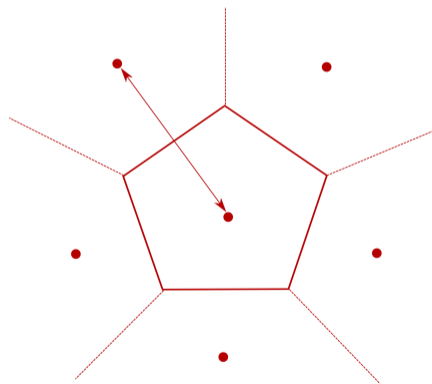


Вычисление градиента: 3D

▶ В 3D ячейки могут иметь множество поверхностей.

▶ Основные методы расчёта градиента в центре ячейки:

1. Green-Gauss Cell Based (linear)
2. Green-Gauss Node Based (pointLinear)
3. Least-Squares (leastSquires)



- ▶ Теорема Гаусса-Остроградского

$$\int_V [\nabla \cdot B] dV = \int_S [B \cdot \hat{n}] dS \quad (46)$$

- ▶ Пусть $B = T \cdot c$ - произведение скаляра на вектор.

$$\int_V [\nabla \cdot (Tc)] dV = \int_S [(Tc) \cdot \hat{n}] dS \quad (47)$$

- ▶ Из скалярного произведения:

$$\int_V [T (\nabla c)] dV + \int_V [c (\nabla T)] dV = \int_S [T (c \cdot \hat{n})] dS + \int_S [c (T \hat{n})] dS \quad (48)$$

► Пусть $c = (1, 1, 1)$

$$\underbrace{\int_V [T (\nabla c)] dV}_{=0} + \int_V [c (\nabla T)] dV = \underbrace{\int_S [T (c \cdot \hat{n})] dS}_{=0} + \int_S [c (T \hat{n})] dS \quad (49)$$

$$\int_V (\nabla T) dV = \int_S (T \hat{n}) dS \quad (50)$$

- ▶ Каждая ячейка имеет конечное число поверхностей (N).
- ▶ Заменяем поверхностный интеграл на сумму интегралов по каждой поверхности:

$$\int_V (\nabla T) dV = \sum_N \int_S (T \hat{n}) dS \quad (51)$$

- ▶ Изменение величины по поверхности линейно!
- ▶ Заменяем поверхностный интеграл на значение величины в центре поверхности:

$$\int_V (\nabla T) dV = \sum_N [T_f \cdot \hat{n}_f \cdot A_f] \quad (52)$$

- ▶ Изменение величины по объёму тоже линейно!
- ▶ Заменяем объёмный интеграл на значение в центре ячейки $(\nabla T)_P$:

$$(\nabla T)_P V_P = \sum_N [T_f \cdot \hat{n}_f \cdot A_f] \quad (53)$$

- ▶ Градиент в центре ячейки для любой величины:

$$(\nabla T)_P = \frac{1}{V_P} \sum_N [T_f \cdot \hat{n}_f \cdot A_f] \quad (54)$$

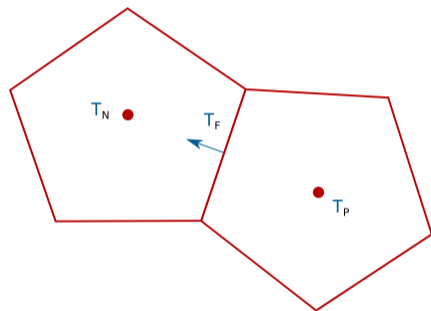
- ▶ Вычисляем значение в центре поверхности T_f .
- ▶ Определяем $\hat{n}_f \cdot A_f$.
- ▶ Определяем \sum_N .
- ▶ Делим на объём V_P .
- ▶ Существует два метода определения T_f :
 1. Линейная интерполяция (Cell Based).
 2. Среднее по узлам (Node Based).

Cell Based

- Линейная интерполяция между ячейкой и соседней с ней ячейкой:

$$T_f = T_P \cdot f + T_N \cdot (1 - f) \quad (55)$$

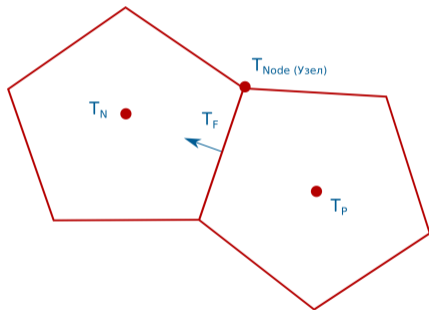
$$f = \frac{|x_f - x_N|}{|x_P - x_N|} \quad (56)$$



Node Based

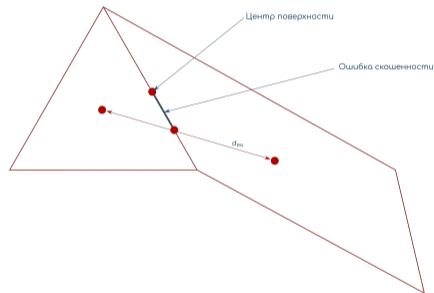
$$T_f = \frac{1}{N} \sum_{Nodes} T_{nodes}$$

(57)



Ошибка скошенности

- ▶ Cell-Based метод рассчитывает T_f на пересечении d_{PN} с поверхностью. Может возникнуть ошибка скошенности.
- ▶ Node-Based метод рассчитывает T_f в центре поверхности.
- ▶ Green-Gauss Node Based и Least Squares не имеют ошибки скошенности.



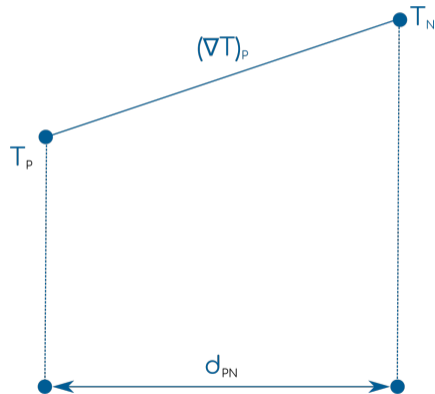
Least-Squares Gradient Scheme

Least-Squares Gradient Scheme

- ▶ Не использует значение на поверхности ячейки T_f .

$$T_N = T_P + d_{PN} \cdot (\nabla T)_P \quad (58)$$

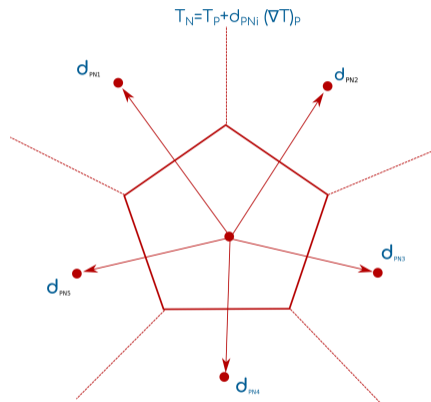
- ▶ $(\nabla T)_P$ - в данном уравнении неизвестно.



- ▶ Запишем дополнительное уравнение для каждой поверхности ячейки:

$$T_{N_i} = T_P + d_{PN_i} \cdot (\nabla T)_P \quad (59)$$

- ▶ Имеется одно уравнение для каждой поверхности ячейки.
- ▶ В каждом уравнении неизвестный $(\nabla T)_P$ один и тот же.
- ▶ d_{PN_i} - расстояние между центрами соседних ячеек.



- ▶ Вычтем значение T_P из обеих частей уравнения:

$$T_{N_i} - T_P = d_{PN_i} \cdot (\nabla T)_P \quad (60)$$

- ▶ Теперь имеется матрица уравнений:

$$[T_N - T_P] = [d_{PN}] [(\nabla T)_P] \quad (61)$$

$$[d_{PN}] [(\nabla T)_P] = [T_N - T_P] \quad (62)$$

$$AX = B \quad (63)$$

$$\begin{vmatrix} d_{PN_1,x} & d_{PN_1,y} & d_{PN_1,z} \\ d_{PN_2,x} & d_{PN_2,y} & d_{PN_2,z} \\ d_{PN_3,x} & d_{PN_3,y} & d_{PN_3,z} \\ d_{PN_4,x} & d_{PN_4,y} & d_{PN_4,z} \end{vmatrix} \cdot \begin{vmatrix} \left(\frac{\partial T}{\partial x}\right)_P \\ \left(\frac{\partial T}{\partial y}\right)_P \\ \left(\frac{\partial T}{\partial z}\right)_P \end{vmatrix} = \begin{vmatrix} T_{N_1} - T_P \\ T_{N_2} - T_P \\ T_{N_3} - T_P \\ T_{N_4} - T_P \end{vmatrix} \quad (64)$$

$$(4 \times 3) \times (3 \times 1) = (4 \times 1) \quad (65)$$

► или в общем виде:

$$(N \times 3) \times (N \times 1) = (N \times 1) \quad (66)$$

$$(N \times 3) \times (N \times 1) = (N \times 1) \quad (67)$$

- ▶ Неквадратная матрица - невозможно получить точное решение.
- ▶ Можно рассчитать приближённое решение для $(\nabla T)_P$.
- ▶ Ошибка вычисления будет равна:

$$e_i = T_{N_i} - (T_P + (\nabla T)_P \cdot d_{PN_i}) \quad (68)$$

- ▶ Least-Squares минимизирует сумму "квадрата ошибок".

$$(\nabla T)_P = (d^T d)^{-1} d^T (T_N - T_P) \quad (69)$$

$$G = (d^T d) \quad (70)$$

$$(\nabla T)_P = G^{-1} d^T (T_N - T_P) \quad (71)$$

$$G = \begin{vmatrix} d_{PN_1,x} & d_{PN_2,x} & d_{PN_3,x} & d_{PN_4,x} \\ d_{PN_1,y} & d_{PN_2,y} & d_{PN_3,y} & d_{PN_4,y} \\ d_{PN_1,z} & d_{PN_2,z} & d_{PN_3,z} & d_{PN_4,z} \end{vmatrix} \cdot \begin{vmatrix} d_{PN_1,x} & d_{PN_1,y} & d_{PN_1,z} \\ d_{PN_2,x} & d_{PN_2,y} & d_{PN_2,z} \\ d_{PN_3,x} & d_{PN_3,y} & d_{PN_3,z} \\ d_{PN_4,x} & d_{PN_4,y} & d_{PN_4,z} \end{vmatrix} \quad (72)$$

$$G = \begin{vmatrix} G_{11} & G_{12} & G_{13} \\ G_{21} & G_{22} & G_{23} \\ G_{31} & G_{32} & G_{33} \end{vmatrix} \quad (73)$$

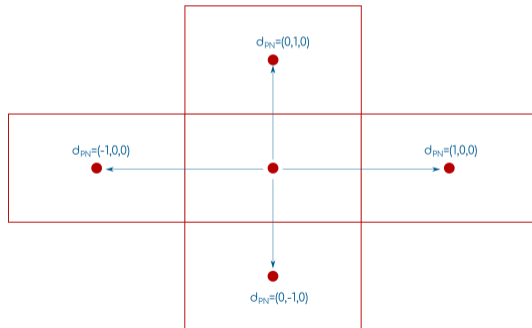
► $(3 \times N) \times (N \times 3) = (3 \times 3) \Rightarrow G$ всегда матрица $3 \times 3 \Rightarrow$ есть G^{-1}

$$(\nabla T)_P = G^{-1} d^T (T_N - T_P) \quad (74)$$

- ▶ Если сетка стационарна, можно рассчитать G^{-1} один раз для каждой ячейки сетки.
- ▶ $(\nabla T)_P$ можно получить из матрицы умножением.

Пример

- ▶ Рассмотрим сетку из прямоугольников.
- ▶ Расстояние между центрами - 1 метр.



Пример

- ▶ Допустим температуры $^{\circ}\text{C}$:

$$\begin{array}{ccc} & 300 & \\ 100 & 200 & 300 \Rightarrow \\ & 100 & \end{array}$$

- ▶ Рассчитаем $(\nabla T)_P$

Пример

$$d = \begin{vmatrix} d_{PN_1} \\ d_{PN_2} \\ d_{PN_3} \\ d_{PN_4} \end{vmatrix} = \begin{vmatrix} -1, & 0, & 0 \\ 0, & -1, & 0 \\ 1, & 0, & 0 \\ 0, & 1, & 0 \end{vmatrix}; \quad d^T = \begin{vmatrix} -1, & 0, & 1, & 0 \\ 0, & -1, & 0, & 1 \\ 0, & 0, & 0, & 0 \end{vmatrix} \Rightarrow$$

$$G = \begin{vmatrix} 2, & 0, & 0 \\ 0, & 2, & 0 \\ 0, & 0, & 0 \end{vmatrix}; \quad G^{-1} = \begin{vmatrix} 1/2, & 0, & 0 \\ 0, & 1/2, & 0 \\ 0, & 0, & 0 \end{vmatrix}$$

$$(\nabla T)_P = G^{-1} d^T (T_N - T_P)$$

$$(\nabla T)_P = \begin{vmatrix} 1/2, & 0, & 0 \\ 0, & 1/2, & 0 \\ 0, & 0, & 0 \end{vmatrix} \cdot \begin{vmatrix} -1, & 0, & 1, & 0 \\ 0, & -1, & 0, & 1 \\ 0, & 0, & 0, & 0 \end{vmatrix} \cdot \begin{vmatrix} 100 - 200 \\ 100 - 200 \\ 300 - 200 \\ 300 - 200 \end{vmatrix}$$

$$(\nabla T)_P = \begin{vmatrix} -1/2, & 0, & 1/2, & 0 \\ 0, & -1/2, & 0, & 1/2 \\ 0, & 0, & 0, & 0 \end{vmatrix} \cdot \begin{vmatrix} -100 \\ -100 \\ 100 \\ 100 \end{vmatrix}$$

$$(\nabla T)_P = (100, 100, 0) \Rightarrow \frac{\partial T}{\partial x} = 100, \frac{\partial T}{\partial y} = 100, \frac{\partial T}{\partial z} = 0$$

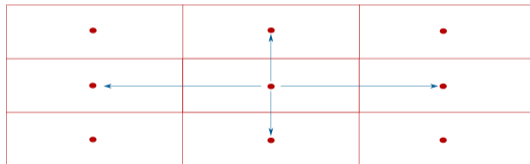
- ▶ Если температура изменилась, то необходимо вычислить градиент снова:

$$(\nabla T)_P = \begin{vmatrix} -1/2, & 0, & 1/2, & 0 \\ 0, & -1/2, & 0, & 1/2 \\ 0, & 0, & 0, & 0 \end{vmatrix} \cdot \begin{vmatrix} T_{N_1} - T_P \\ T_{N_2} - T_P \\ T_{N_3} - T_P \\ T_{N_4} - T_P \end{vmatrix}$$

- ▶ Для статической сетки $G^{-1}d^T$ не изменяется.

Ячейки с большим аспектным соотношением

- ▶ Для тонких ячеек пограничного слоя, расстояние по нормали к стенке d мало.
- ▶ \Rightarrow в Least-Squares методе в градиент основной вклад вносит градиент по потоку.



Весовая функция

$$w = 1/|d| \quad (75)$$

$$\begin{aligned} [d_{PN}] [(\nabla T)_P] &= [T_N - T_P] \\ [w] [d_{PN}] [(\nabla T)_P] &= [w] [T_N - T_P] \end{aligned}$$

- ▶ w - диагональная матрица весов ($w_i = 1/|d_i|$).
- ▶ Теперь решение имеет вид:

$$(\nabla T)_P = G^{-1} d^T w^T w (T_N - T_P), \quad G = [d^T w^T w d] \quad (76)$$

- ▶ Least-Squares метод не использует теорему о дивергенции.
- ▶ Необходима весовая функция для Least-Squares для расчёта на тонких ячейках в пограничном слое.
- ▶ Нужно один раз оценить G^{-1} для статических сеток. Метод достаточно эффективен для расчётов.