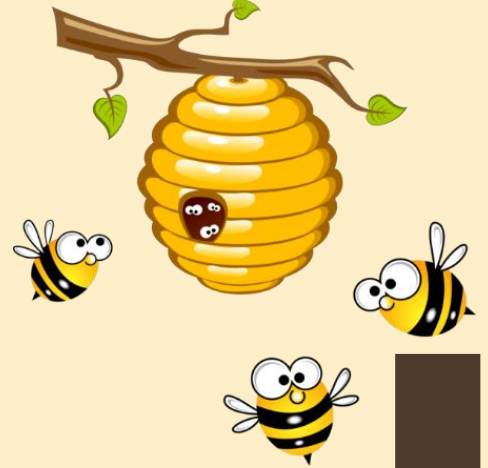




О МЕТОДАХ РОЕВОГО ИНТЕЛЛЕКТА В ОПТИМИЗАЦИИ ПАРАМЕТРОВ ОПЕРАЦИИ ПО ФОРМИРОВАНИЮ СОСУДИСТЫХ АНАСТОМОЗОВ

Куянова Ю. О¹., Дубовой А. В.²., Бервицкий А. В.²., Паршин Д. В.¹

1) Институт гидродинамики им. М.А. Лаврентьева СО РАН

2) ФГБУ "Федеральный центр нейрохирургии"

Виртуальный анастомоз

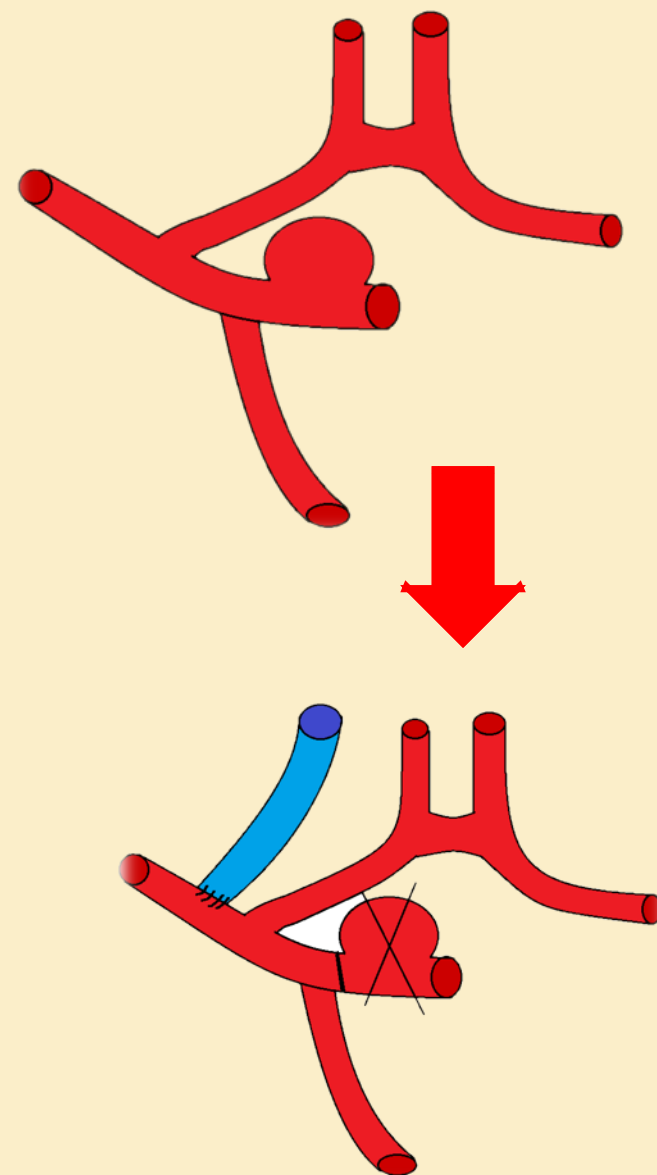
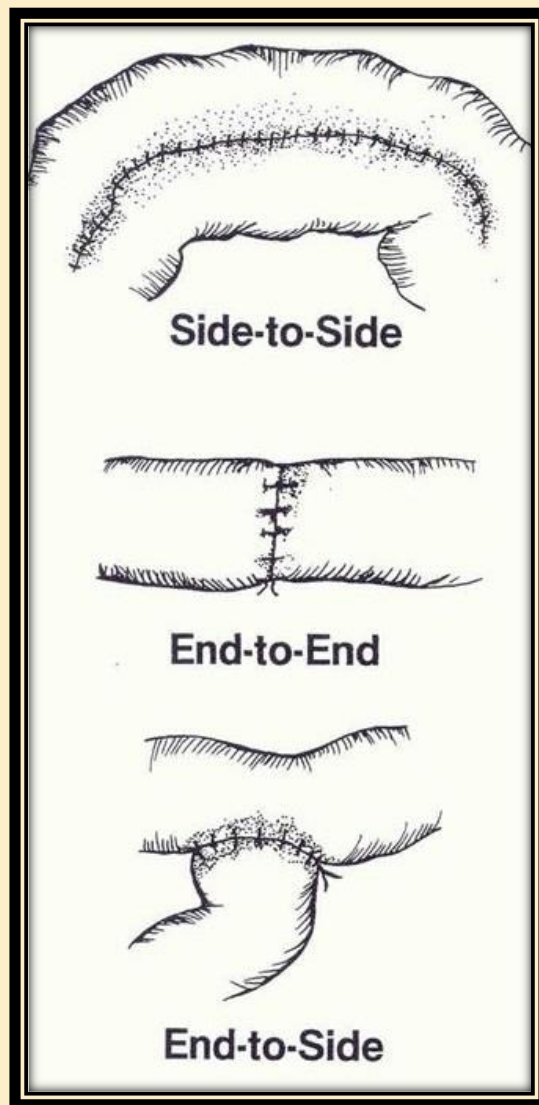
➤ Церебральный анастомоз- это искусственно восстановленное или созданное кровеносное русло, компенсирующее недостающий объем кровотока.

➤ Применение:

- Лечение церебральных аневризм
- Стеноз церебральных артерий
- Опухоли
- Болезнь Мойя-Мойя

➤ Типы анастомозов:

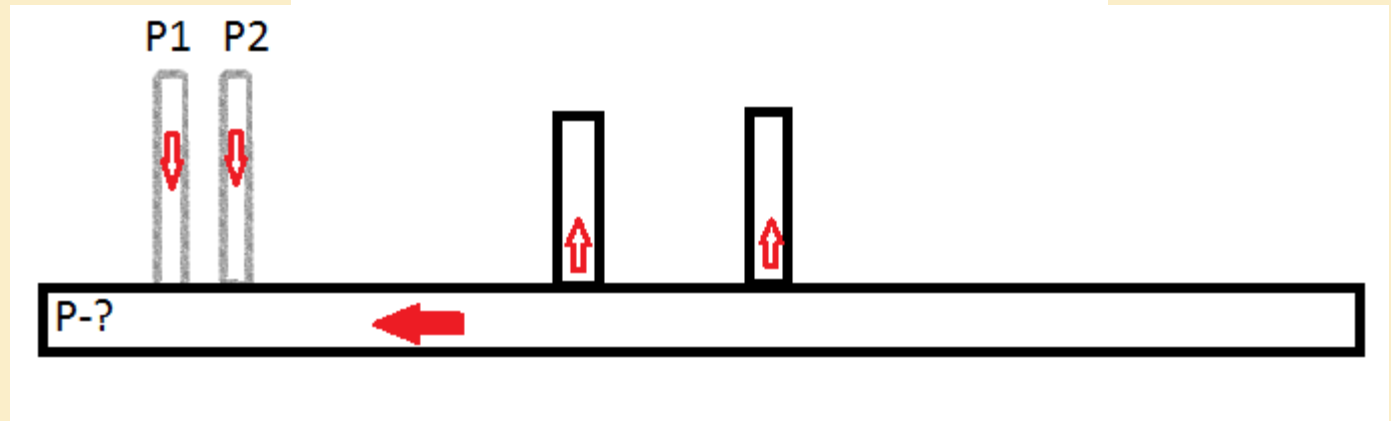
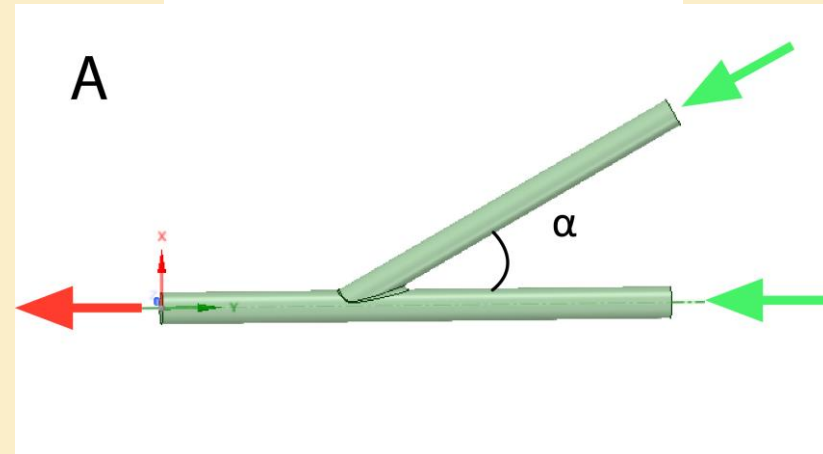
1. Бок-в-бок
2. Конец-в-конец
3. Конец-в-бок



Возникающие задачи

- Необходимость установки анастомоза
- Оптимизация материала артерии-донора
- Оптимизация угла установки
- Оптимизация места установки

TO BE OR NOT TO BE?
That is the question.



Модель электрической цепи

■ Гемодинамические параметры можно интерпретировать следующим образом:

- P (давление) – напряжение в электрической цепи
- Q (объёмный расход) – ток в цепи
- C (отношение изменения объема сети при заданном давлении) – ёмкость конденсатора

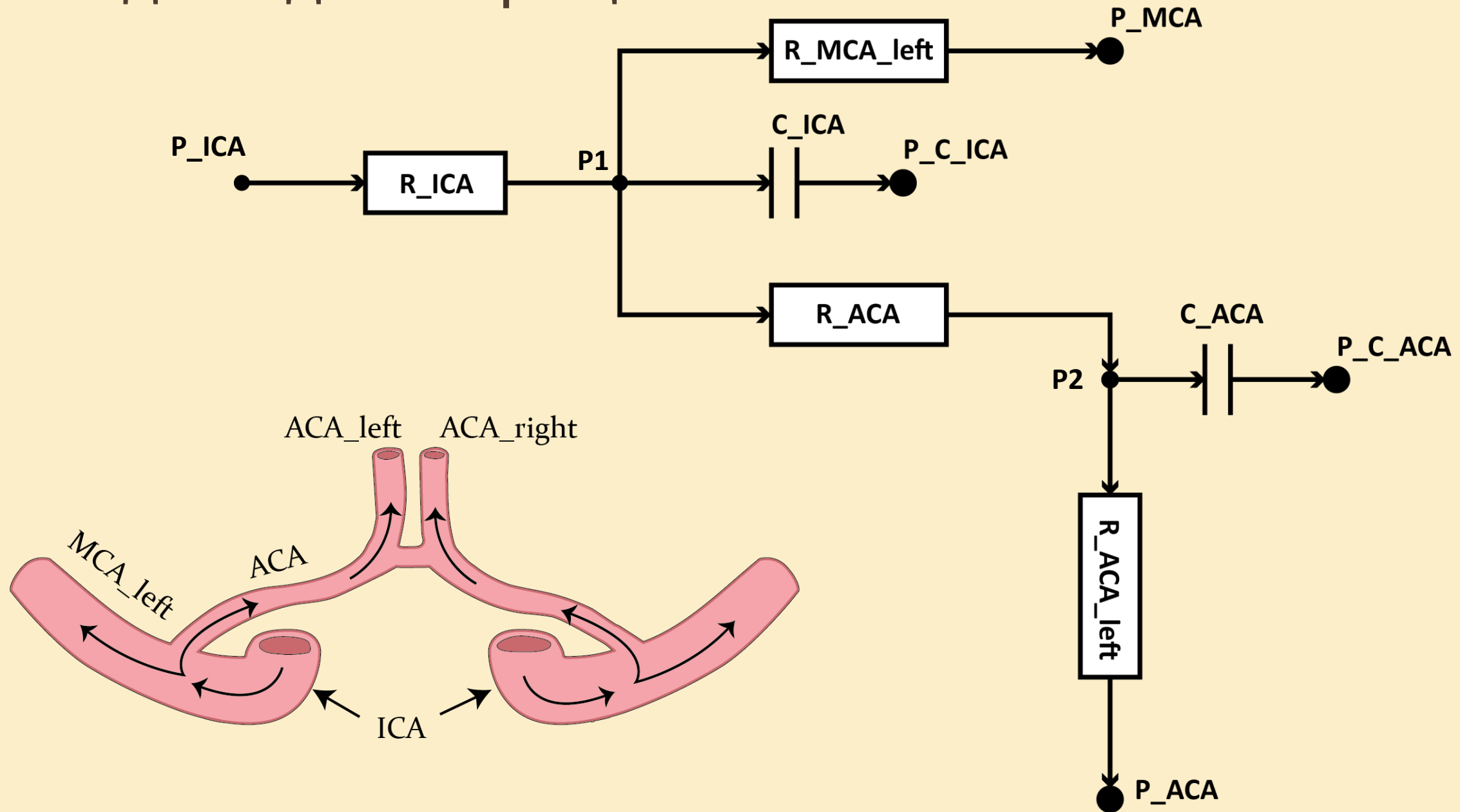
Для вычисления параметров использовались следующие формулы:

$$R = \frac{8\mu l}{\pi r^4}, C = \frac{3}{2}\pi r^2 \frac{r}{h} \frac{1}{E}$$
$$\Delta P = QR,$$

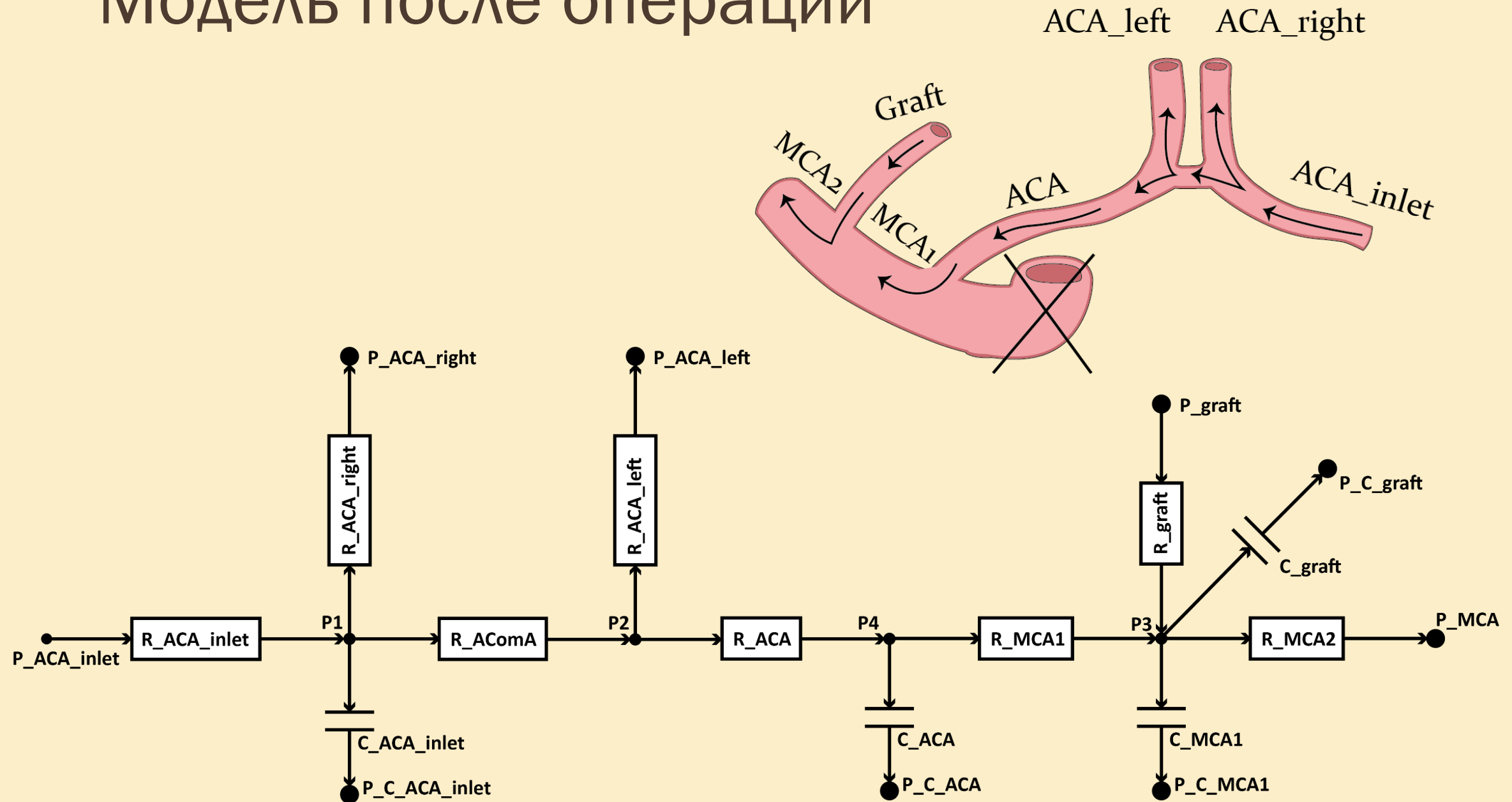
где μ – динамическая вязкость крови, l, r – длина и радиус сосуда, $E \in (8, 16) \text{ dyn/cm}^2$ – модуль Юнга стенки сосуда, $\frac{r}{h} \in (8, 10)$ – коэффициент толщины стенки.

В расчётах использовались значения $E = 10 \text{ dyn/cm}^2, \frac{r}{h} = 10$.

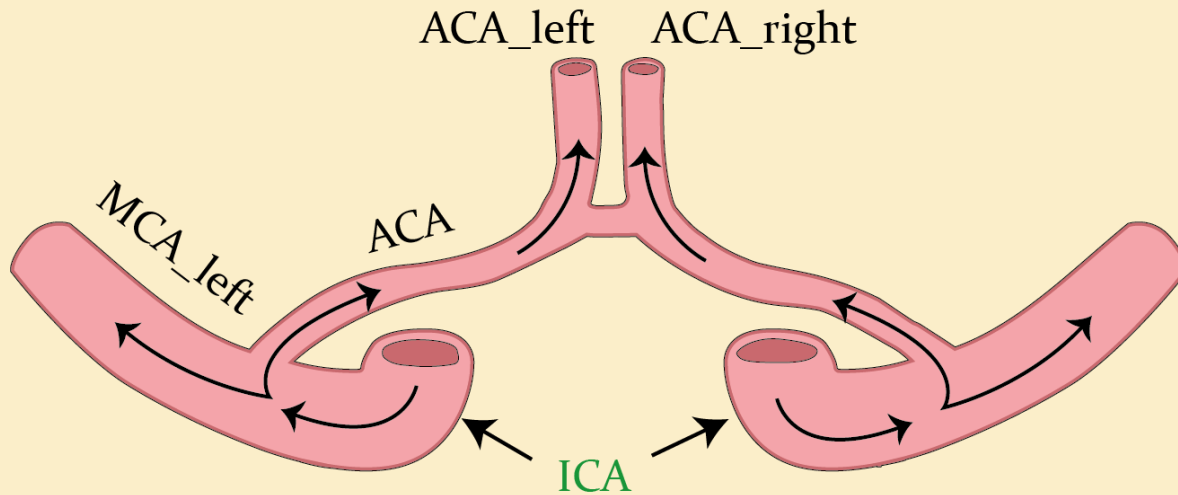
Модель до операции



Модель после операции



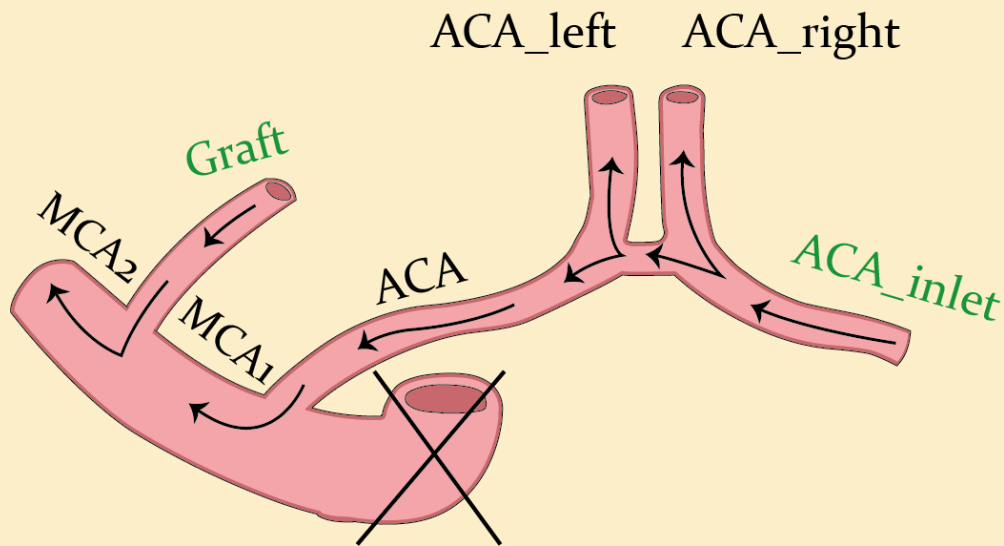
Модель электрической цепи



$$P1 = P_{ICA} - Q_{ICA} \cdot R_{ICA}; P_{MCA} = P2 = P_{C_{ACA}}$$

$$P_{MCA} = P1 - Q_{MCA} \cdot R_{MCA},$$

$$Q_{MCA} = \frac{Q_{ICA}}{1 + R_{MCA} \cdot (C_{ICA} \cdot \omega \cdot i + \frac{1}{R_{ACA}})}$$



$$P1 = P_{ACA_{in}} - Q_{ACA_{in}} \cdot R_{ACA_{in}};$$

$$Q_{AComa} = Q_{ICA} / (1 + R_{AComa} \cdot (C_{ACA_{in}} \cdot \omega \cdot i + 1/R_{ACA_r}))$$

$$P_{ACA_r} = P2 = P1 - Q_{AComa} \cdot R_{AComa}$$

$$Q_{ACA} = Q_{AComa} / (1 + R_{ACA} / R_{ACA_l})$$

$$P4 = P2 - Q_{ACA} \cdot R_{ACA}$$

$$Q_{MCA1} = Q_{ACA} / (1 + R_{MCA1} \cdot C_{ACA} \cdot \omega \cdot i)$$

$$P3 = P4 - Q_{MCA1} \cdot R_{MCA1}$$

$$Q_{MCA2} = \frac{Q_{MCA1} + Q_{graft}}{\omega \cdot i (C_{MCA1} + C_{graft}) \cdot R_{MCA2}}$$

$$P_{MCA} = P3 - Q_{MCA2} \cdot R_{MCA2},$$

Роевой интеллект

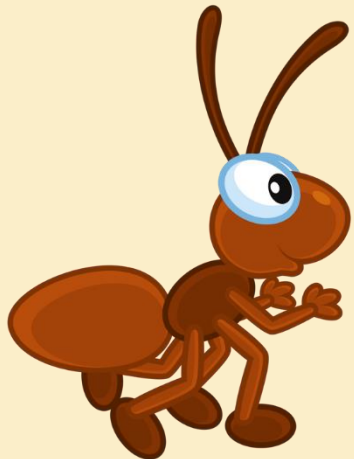
Херардо Бени, Ван Цзин, 1989 г.



- Коллективное поведение децентрализованной самоорганизующейся системы. Рассматривается в теории искусственного интеллекта как метод оптимизации.

Для системы характерно следующее:

- Состоит из множества агентов (боидов).
- Отсутствует централизованная система управления поведения
- Каждый боид следует очень простым правилам
- Случайные взаимодействия приводят к возникновению интеллектуального группового поведения, неконтролируемого отдельными боидами.



Метод роя частиц. Кеннеди, Эберхарт, 1995 г. [1]

Метод численной оптимизации, для использования которого не требуется знать точного градиента оптимизируемой функции. МРЧ оптимизирует функцию, поддерживая популяцию возможных решений, называемых частицами, и перемещая эти частицы в пространстве решений согласно простой формуле.

- ❖ Изначально частицы разбросаны случайным образом по области поиска и каждая имеет случайный вектор скорости;
- ❖ В каждой точке, где побывала каждая частица рассчитывается значение целевой функции;
- ❖ Каждая частица запоминает, какое (и где) лучшее значение целевой функции она лично нашла, а также где расположена точка, являющаяся лучшей для роя в целом;
- ❖ На каждой итерации частица корректирует свою скорость (модуль и направление), чтобы с одной стороны быть поближе к лучшей точке, которую нашла сама, и, в то же время, приблизиться к точке, которая в данный момент является глобально лучшей.

Метод роя частиц. Алгоритм

Постановка задачи:

Пусть $f: R^n \rightarrow R$ – целевая функция;

S – количество частиц в рое;

$x_i \in R^n$ – координата i -ой частицы роя;

$v_i \in R^n$ – скорость i -ой частицы роя;

p_i – лучшее из известных положений i -ой частицы;

g – наилучшее известное состояние роя в целом;

Метод роя частиц. Алгоритм. Шаг 1

Для каждой частицы $i=1,\dots,S$ необходимо:

- Сгенерировать начальное положение частицы с помощью случайного вектора $\mathbf{x}_i \sim U(\mathbf{b}_{lo}, \mathbf{b}_{up})$, имеющего многомерное равномерное распределение. $\mathbf{b}_{lo}$ и $\mathbf{b}_{up}$ — нижняя и верхняя границы пространства решений соответственно.
- Присвоить лучшему известному положению частицы его начальное значение: $\mathbf{p}_i \leftarrow \mathbf{x}_i$.
- Если $(f(\mathbf{p}_i) < f(\mathbf{g}))$, то обновить наилучшее известное состояние роя: $\mathbf{g} \leftarrow \mathbf{p}_i$.
- Присвоить значение скорости частицы: $\mathbf{v}_i \sim U(-(\mathbf{b}_{up}-\mathbf{b}_{lo}), (\mathbf{b}_{up}-\mathbf{b}_{lo}))$.

Метод роя частиц. Алгоритм. Шаг 2

Пока не выполнен критерий остановки (например, достижение заданного числа итераций или необходимого значения целевой функции), повторять Для каждой частицы $i=1,...,S$:

- Сгенерировать случайные векторы $\mathbf{r}_p, \mathbf{r}_g \sim U(0,1)$.
- Обновить скорость частицы: $\mathbf{v}_i \leftarrow \omega \mathbf{v}_i + \phi_p \mathbf{r}_p \times (\mathbf{p}_i - \mathbf{x}_i) + \phi_g \mathbf{r}_g \times (\mathbf{g} - \mathbf{x}_i)$, где операция $\times$ означает покомпонентное умножение.
- Обновить положение частицы переносом $\mathbf{x}_i$ на вектор скорости: $\mathbf{x}_i \leftarrow \mathbf{x}_i + \mathbf{v}_i$. Заметим, что этот шаг выполняется вне зависимости от улучшения значения целевой функции.
- Если $(f(\mathbf{x}_i) < f(\mathbf{p}_i))$, то делать:
 - Обновить лучшее известное положение частицы: $\mathbf{p}_i \leftarrow \mathbf{x}_i$.
 - Если $(f(\mathbf{p}_i) < f(\mathbf{g}))$, то обновить лучшее известное состояние роя в целом: $\mathbf{g} \leftarrow \mathbf{p}_i$.

Теперь $\mathbf{g}$ содержит наилучшее из
найденных решений

Метод роя частиц. Канонический алгоритм

Что важно: от коррекции скорости зависит сходимость алгоритма!

$$\mathbf{v}_i \leftarrow \omega \mathbf{v}_i + \varphi_p \mathbf{r}_p \times (\mathbf{p}_i - \mathbf{x}_i) + \varphi_g \mathbf{r}_g \times (\mathbf{g} - \mathbf{x}_i),$$

r_p, r_g — случайные числа из интервала (0,1);

φ_p, φ_g — веса, подбираемые под конкретную задачу;

$\omega(t)$ — коэффициент для более плавного изменения скорости

- Один из самых распространенных вариантов алгоритма вводит нормировку коэффициентов φ_p и φ_g , чтобы сходимость не так сильно зависела от их выбора:

$$\mathbf{v}_i \leftarrow \chi(\omega \mathbf{v}_i + \varphi_p \mathbf{r}_p \times (\mathbf{p}_i - \mathbf{x}_i) + \varphi_g \mathbf{r}_g \times (\mathbf{g} - \mathbf{x}_i)),$$

$$\chi = \frac{2k}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|},$$

$$\varphi = \varphi_p + \varphi_g,$$

$$\varphi > 4.$$

коэффициент k должен лежать в интервале (0, 1).

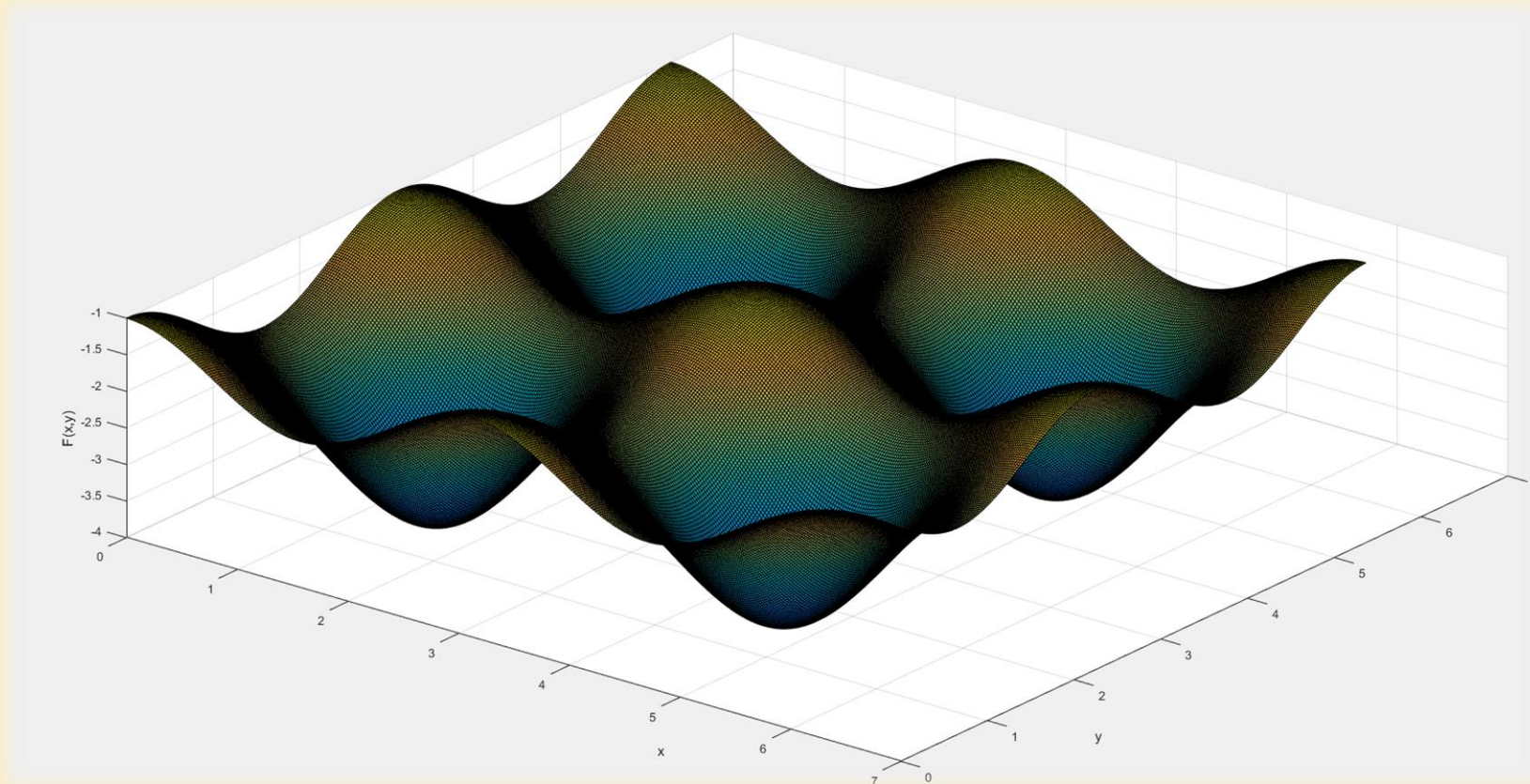
Результаты

- Использовалась каноническая модель
- Были взяты следующие параметры:
 - $\varphi_p = \varphi_g = 1.49618$, $\omega = 0.7298$, $N=5$
 - Критерий остановки – $P2_{MCA} \geq 0.85 \cdot P1_{MCA}$, где $P2_{MCA}$ - давление после операции, $P1_{MCA}$ – до.

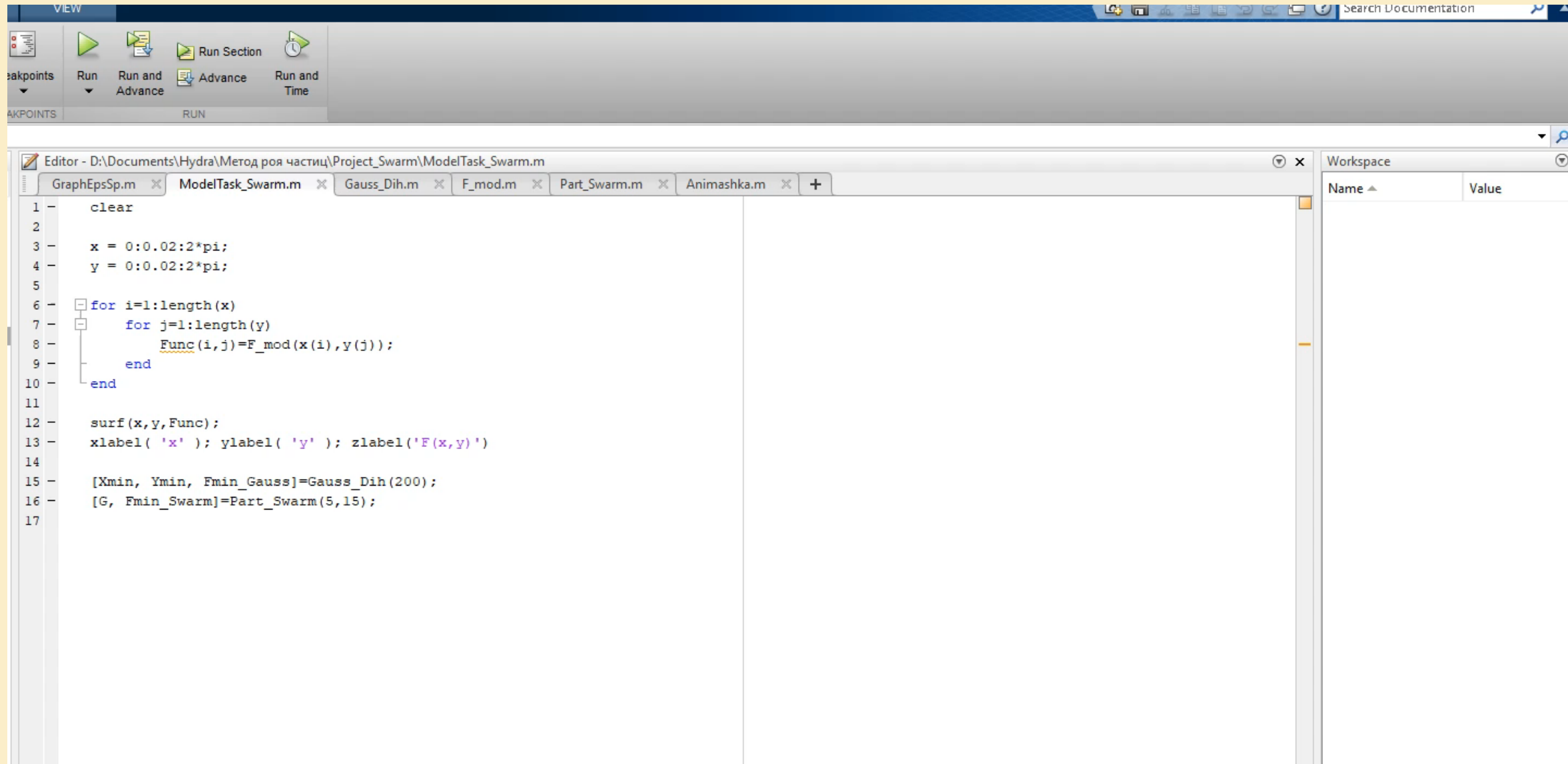
Радиус графта, см	Давление P_{MCA} , mmHg	Длина MCA1, см	Длина графта, см	Количество итераций
$2/3 * r_{MCA1}$	81.31	1.87	14.67	0
$3/4 * r_{MCA1}$	81.04	3.63	12.21	0
r_{MCA1}	77.14	1.36	13.67	0
$4/3 * r_{MCA1}$	73.93	4.89	10.98	1
$3/2 * r_{MCA1}$	72.68	5	14.48	4

Результаты на модельной задаче

- $F(x, y) = -(1 + \sin^2(x)) * (1 + \sin^2(y))$
- $N=5$
- Критерий остановки – выполнение 10-ти итераций



Результаты на модельной задаче



Список литературы

- 1) James Kennedy and Russell Eberhart, "Particle swarm optimization", IEEE, 1995 г.
- 2) Riccardo Poli, James Kennedy, Tim Blackwell, «Particle swarm optimization: an overview», Swarm Intelligence, 2007 г.
- 3) Ding, Jinjin & Wang, Qunjin & Zhang, Qian & Ye, Qiubo & Ma, Yuan. (2019). A Hybrid Particle Swarm Optimization-Cuckoo Search Algorithm and Its Engineering Applications. Mathematical Problems in Engineering. 2019. 1-12. 10.1155/2019/5213759.

Спасибо за внимание!

Контактные данные:

Паршин Даниил Васильевич - danilo.skiman@gmail.com

Куянова Юлия Олеговна – july9696@mail.ru