

# Система для распределенного запуска задач многовариантного анализа MultiGRID

---

*Каргин А.А. (МГТУ им. Н.Э. Баумана, 6-й курс)*

Лапшин А.Е. (МГТУ им. Н.Э. Баумана, 6-й курс)

к.т.н. Овчинников В.А. (ООО «Ладуга»)

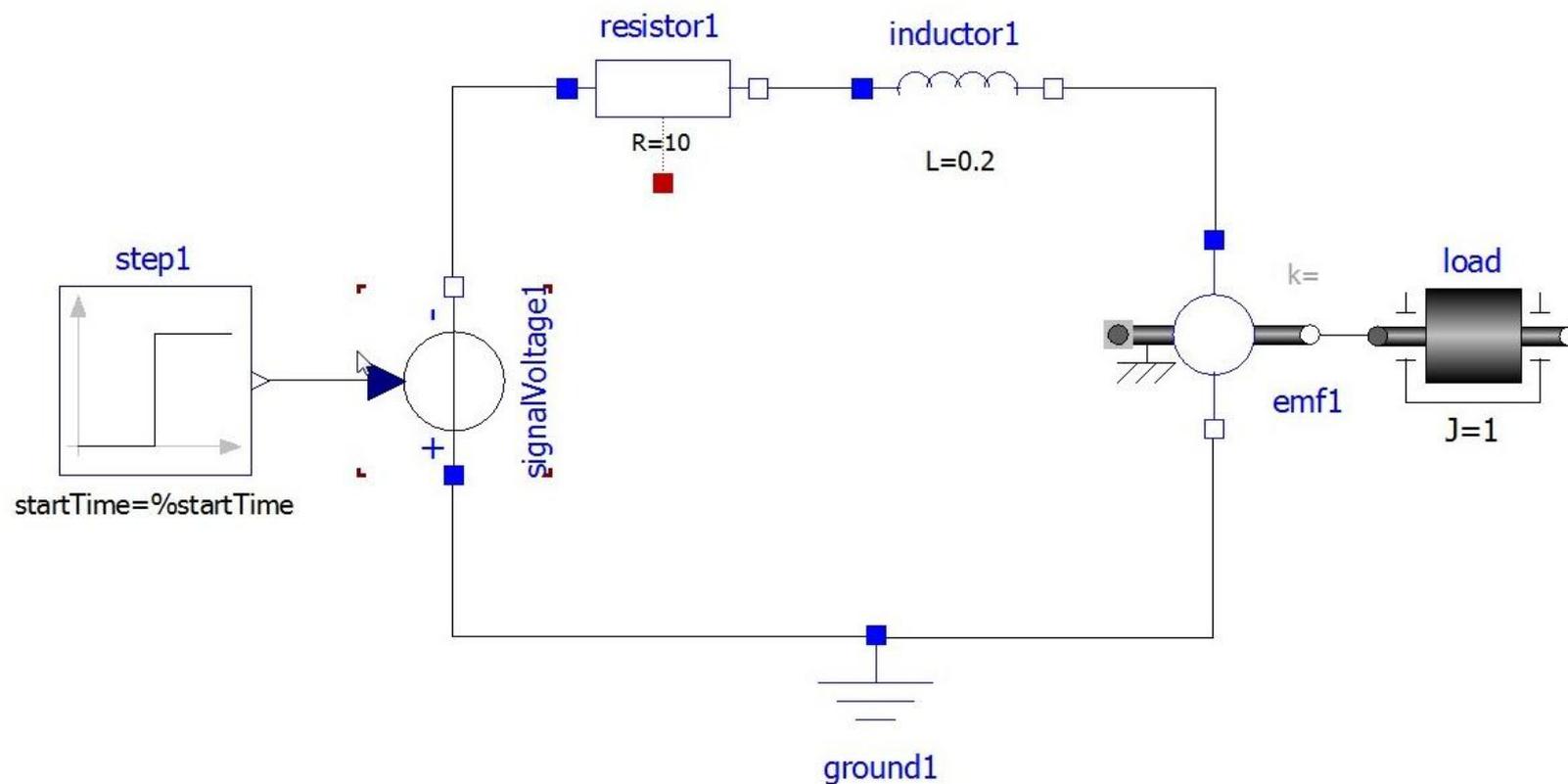
# Язык моделирования Modelica



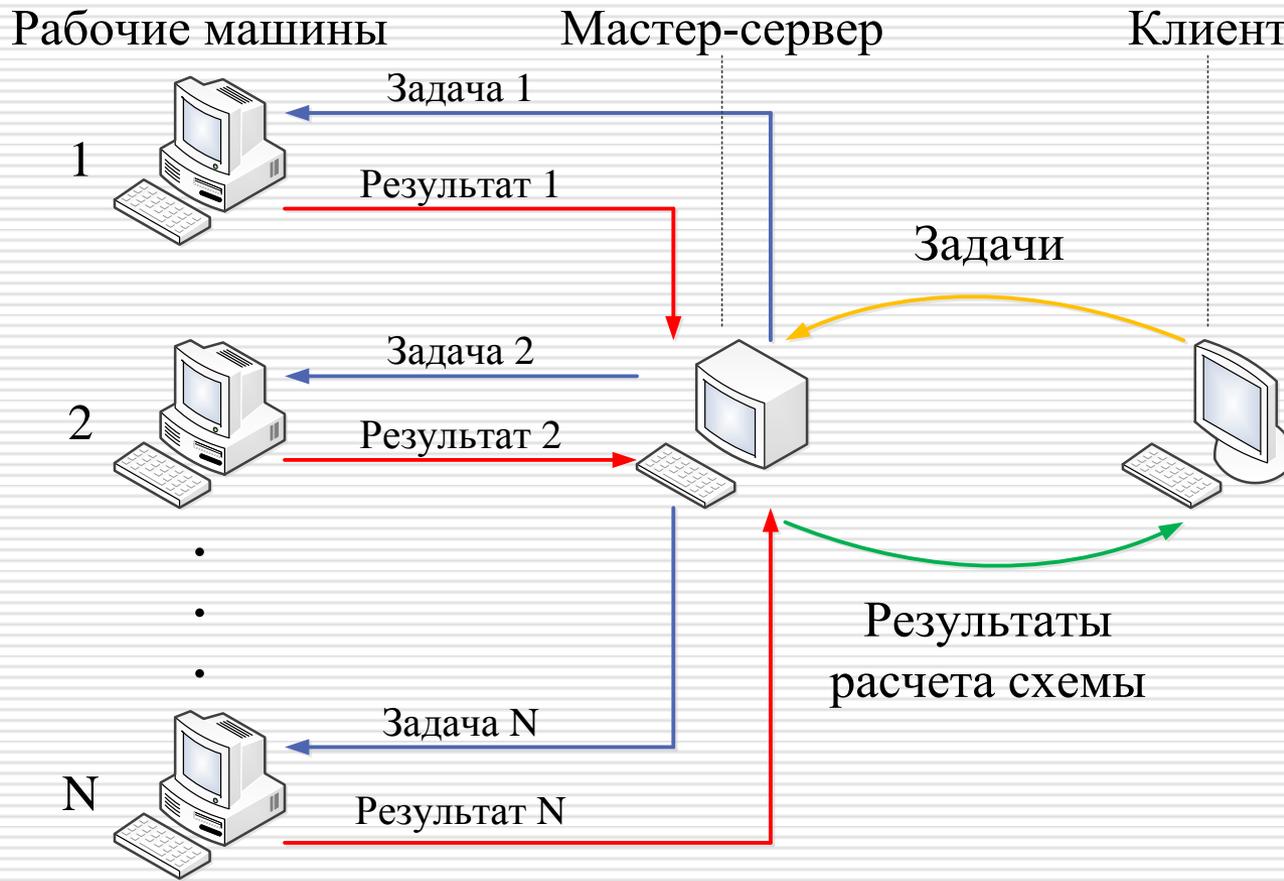
Modelica — это объектно-ориентированный, декларативный, многоцелевой язык описания для компонентно-ориентированного моделирования гибридных систем.

Свободный язык Modelica разработан некоммерческим объединением Modelica Association, которое также разрабатывает бесплатную стандартную библиотеку Modelica, которая содержит около 1360 обобщенных моделей компонентов и 1280 функций в различных областях, согласно версии 3.2.1.

# Модель двигателя постоянного тока, выполненная в OpenModelica Connection Editor



# Общая архитектура GRID-системы



# Требования к балансировщику

---

- открытый исходный код;
- кроссплатформенность;
- написан на C/C++ или на Python;
- невысокие требования к системе
- легкость в установке;
- автоматическая отправка задач рабочим машинам и сбор результатов;
- расширение функционала за счет внешних модулей.



# Celery

---

Это простая, гибкая и надежная распределенная система, служащая для обработки большого количества заданий.

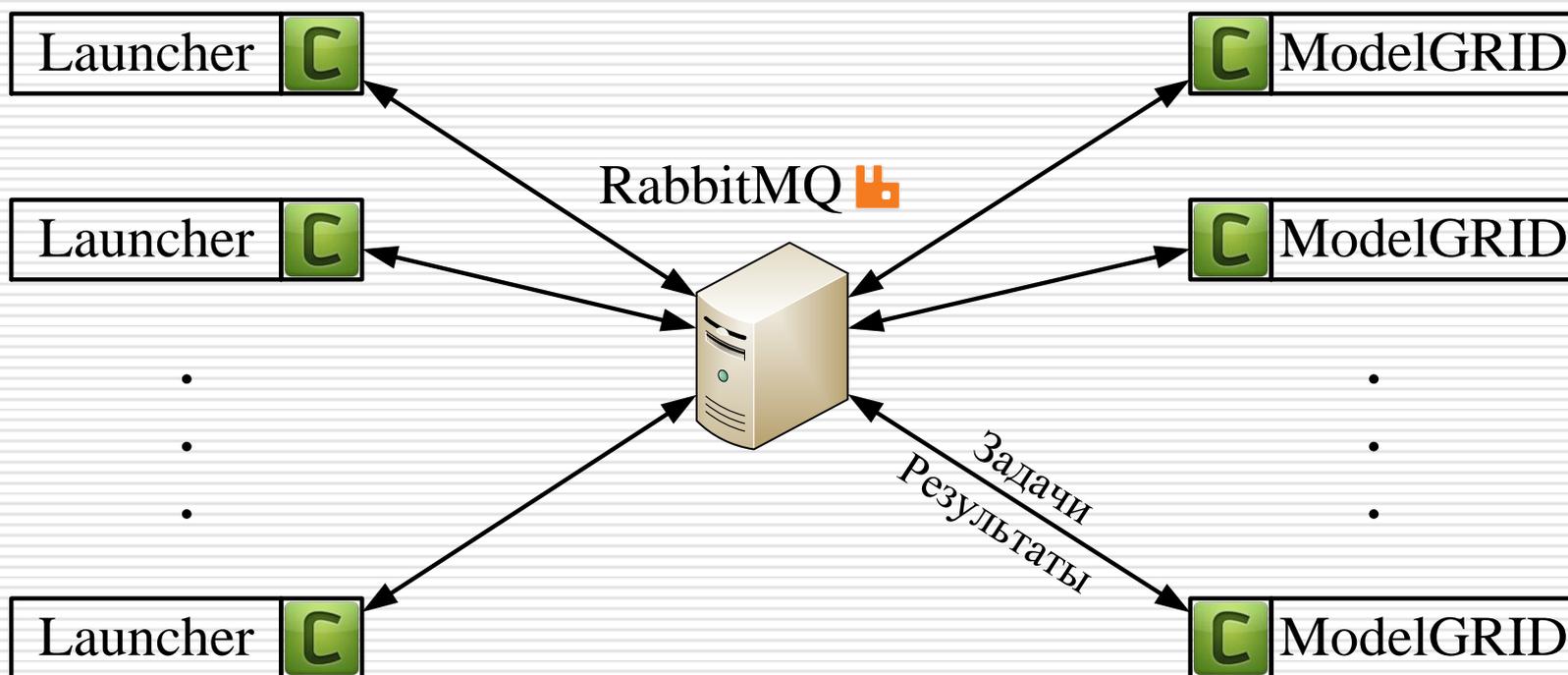
Она проста в использовании и позволяет сконцентрироваться на описании и решении самих задач.

# Реализованная архитектура

---

Рабочие машины

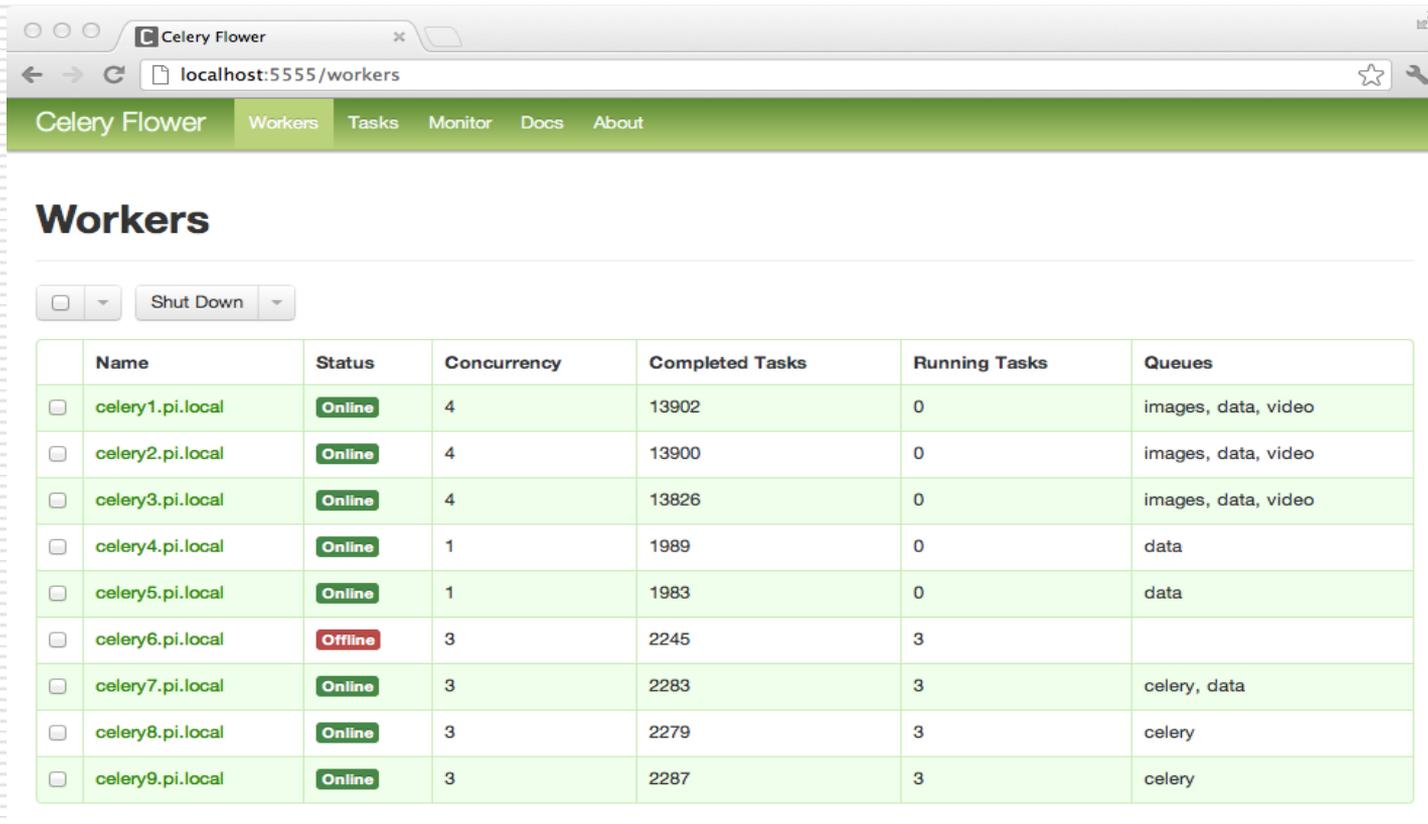
Клиенты



# Мониторинг в Celery



# Администрирование в Celery



The screenshot shows the Celery Flower web interface in a browser window. The address bar displays 'localhost:5555/workers'. The navigation menu includes 'Celery Flower', 'Workers', 'Tasks', 'Monitor', 'Docs', and 'About'. The main heading is 'Workers'. Below the heading, there is a control bar with a checkbox and a 'Shut Down' button. The main content is a table listing the status of various Celery workers.

	Name	Status	Concurrency	Completed Tasks	Running Tasks	Queues
<input type="checkbox"/>	celery1.pi.local	Online	4	13902	0	images, data, video
<input type="checkbox"/>	celery2.pi.local	Online	4	13900	0	images, data, video
<input type="checkbox"/>	celery3.pi.local	Online	4	13826	0	images, data, video
<input type="checkbox"/>	celery4.pi.local	Online	1	1989	0	data
<input type="checkbox"/>	celery5.pi.local	Online	1	1983	0	data
<input type="checkbox"/>	celery6.pi.local	Offline	3	2245	3	
<input type="checkbox"/>	celery7.pi.local	Online	3	2283	3	celery, data
<input type="checkbox"/>	celery8.pi.local	Online	3	2279	3	celery
<input type="checkbox"/>	celery9.pi.local	Online	3	2287	3	celery

# Решатель Modelica отвечает за:

---

1. генерацию необходимых файлов для компиляции модели в исполняемый файл;
2. получение исполняемого файла;
3. запуск исполнимого файла с различными параметрами;
4. чтение и запись результатов расчета.

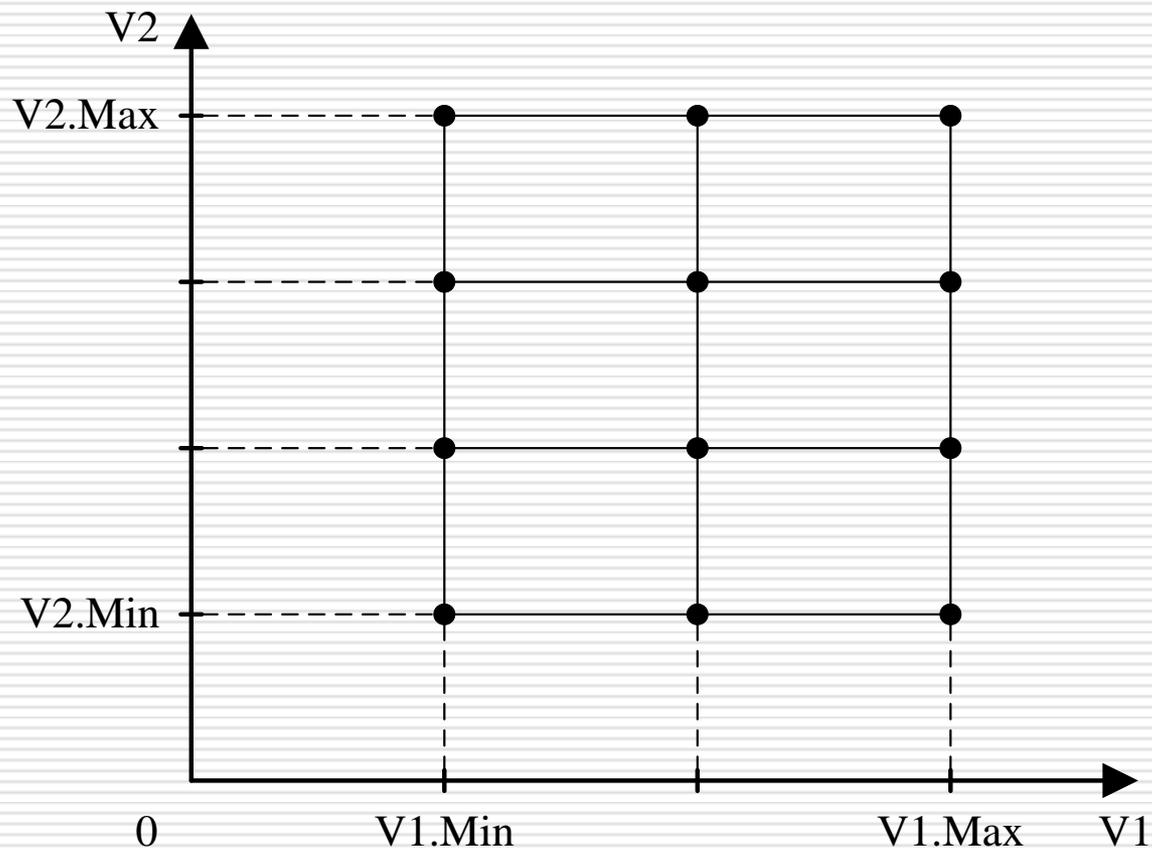
# Этапы получения результатов расчета модели

---



# Метод сканирования

---



# Метод анализа чувствительности

---

## 1. Forward

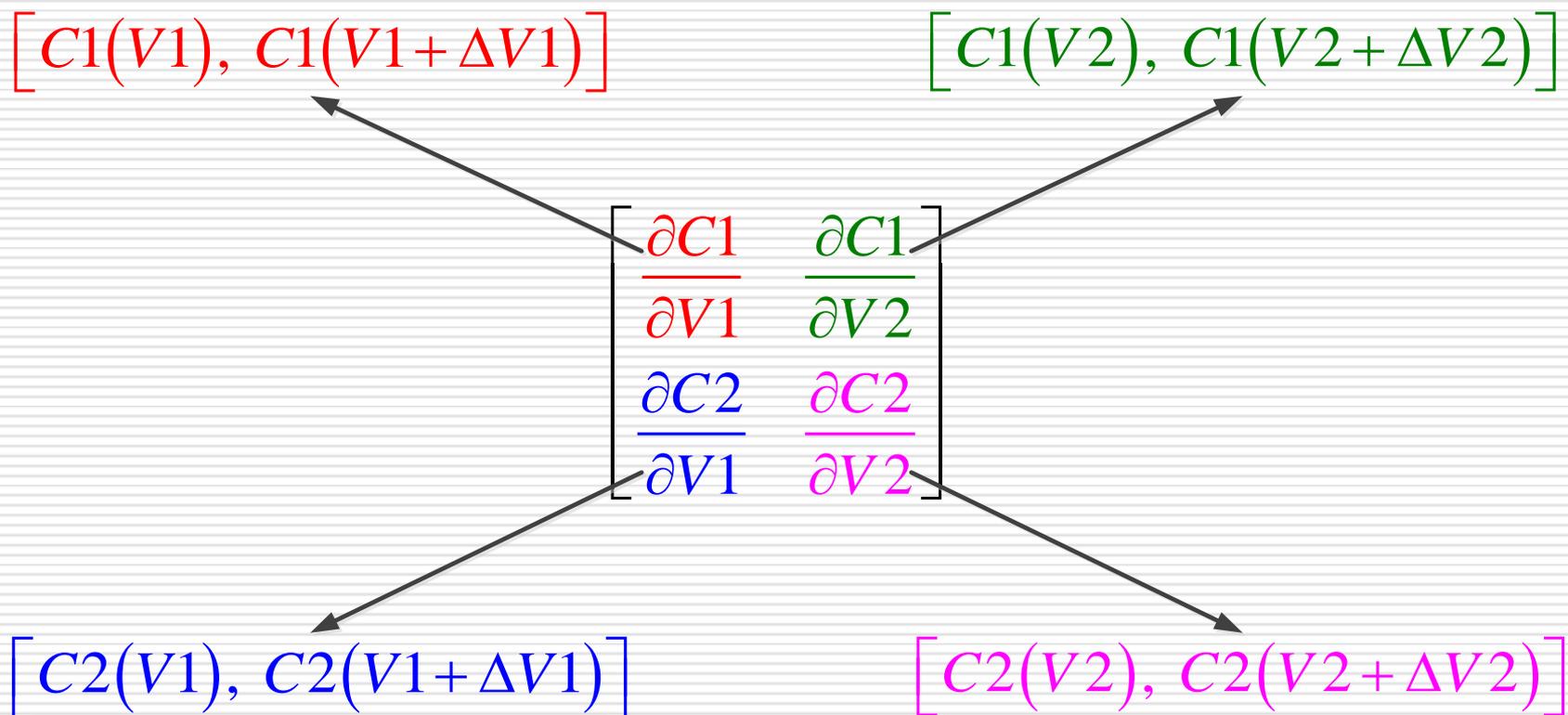
$$\frac{\partial f}{\partial x} = \frac{f(x + \Delta x) - f(x)}{\Delta x} \quad (1)$$

## 2. Backward

$$\frac{\partial f}{\partial x} = \frac{f(x) - f(x - \Delta x)}{\Delta x} \quad (2)$$

# Метод анализа чувствительности

---



# Направления дальнейшей работы

---

1. Реализация поддержки графического пользовательского интерфейса (GUI) программы анализа динамических систем PRADIS;
2. Создание web-интерфейса для задания исходных данных задач;
3. Разработка новых методов многовариантного анализа, использующих преимущества распределенных вычислений;
4. Использование пост- и препроцессоров PRADIS;
5. Разработка новых объектов поддержки решателей.