Инструкции: “Практическая часть”

Кирилл Терехов

https://dodo.inm.ras.ru/terekhov/lect1/prak/presentation.pdf

4 марта 2024 г.

Содержание

Обзор

Репозиторий кода

Пример (Ссылка)

https://github.com/kirill-terekhov/mipt-solvers

Репозиторий кода

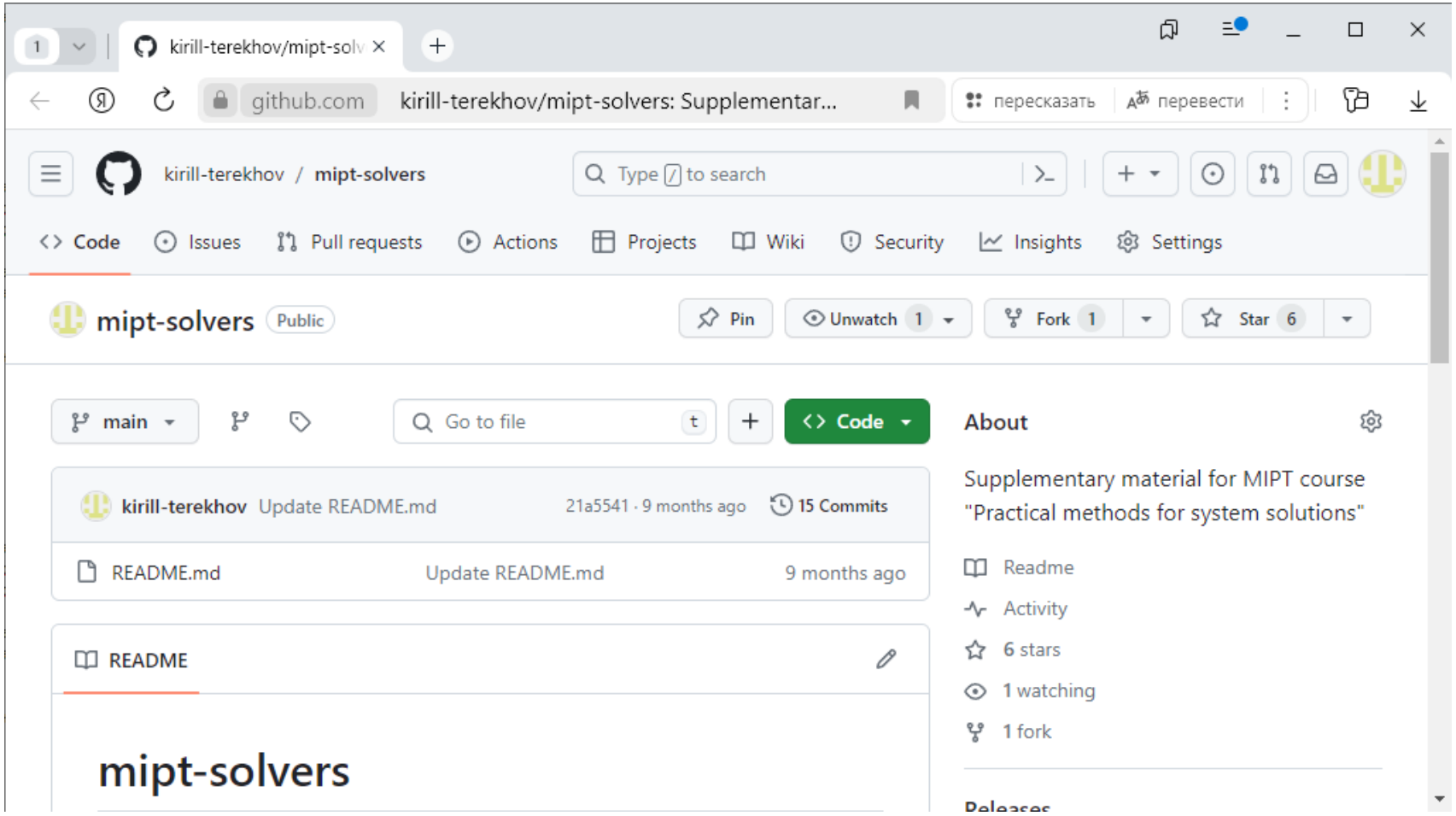

Рис. 1: Сайт репозитория

Содержимое в ветках репозитория

lect0 Примеры систем линейных уравнений и генераторов систем.

lect1 Метод сопряженных градиентов и предобуславливатель Чебышева.

lect2 Метод неполной факторизации Краута с оценкой числа обусловленности обратных факторов.

lect3 Пример геометрического многосеточного метода в одном измерении.

lect4 Алгебраический многосеточный метод Руге-Штюбена со сглаживателем Чебышева.

lect5 Алгебраический многосеточный метод Руге-Штюбена со сглаживателем Якоби и Гаусса-Зейделя.

lect6 S-версия многоуровнего метода неполной факторизации.

Содержимое в ветках репозитория

lect7 M-версия многоуровнего метода неполной факторизации.

lect8 Перемасштабирование и переупорядочивание в методах неполной факторизации.

lect9 Перемасштабирование и переупорядочивание в S и M версиях многоуровневой факторизации.

lect10 Метод сопряженных градиентов Брамбла-Пасьяка для седловых систем.

lect11 Многошаговые методы для задач фильтрации и пороупругости.

lect12 Пример использования метода Ньютона и метода CPR-AMG для решения задачи двухфазного смешиваемого вытеснения.

lect13 Метод дефляций для поиска многих решений в нелинейной задаче Карриера.

Примеры систем

Содержимое ветки lect0, готовые системы в формате *.mtx*:

poisson_10 Задача Пуассона на регулярной сетке $10 \times 10 \times 10$.

poisson_40 Задача Пуассона на регулярной сетке $40 \times 40 \times 40$.

two_wells_tpfa Задача анизотропной диффузии с двумя скважинами, дискретизация МКО с двухточечной аппроксимацией потока.

two_wells_mfd Задача анизотропной диффузии с двумя скважинами, дискретизация методом опорных операторов.

two_wells_saddle_darcy Задача анизотропной диффузии с двумя скважинами, дискретизация системы в седловой форме.

norne_tpfa Задача анизотропной диффузии с использованием данных месторождения Норна.

Иллюстрации примеров

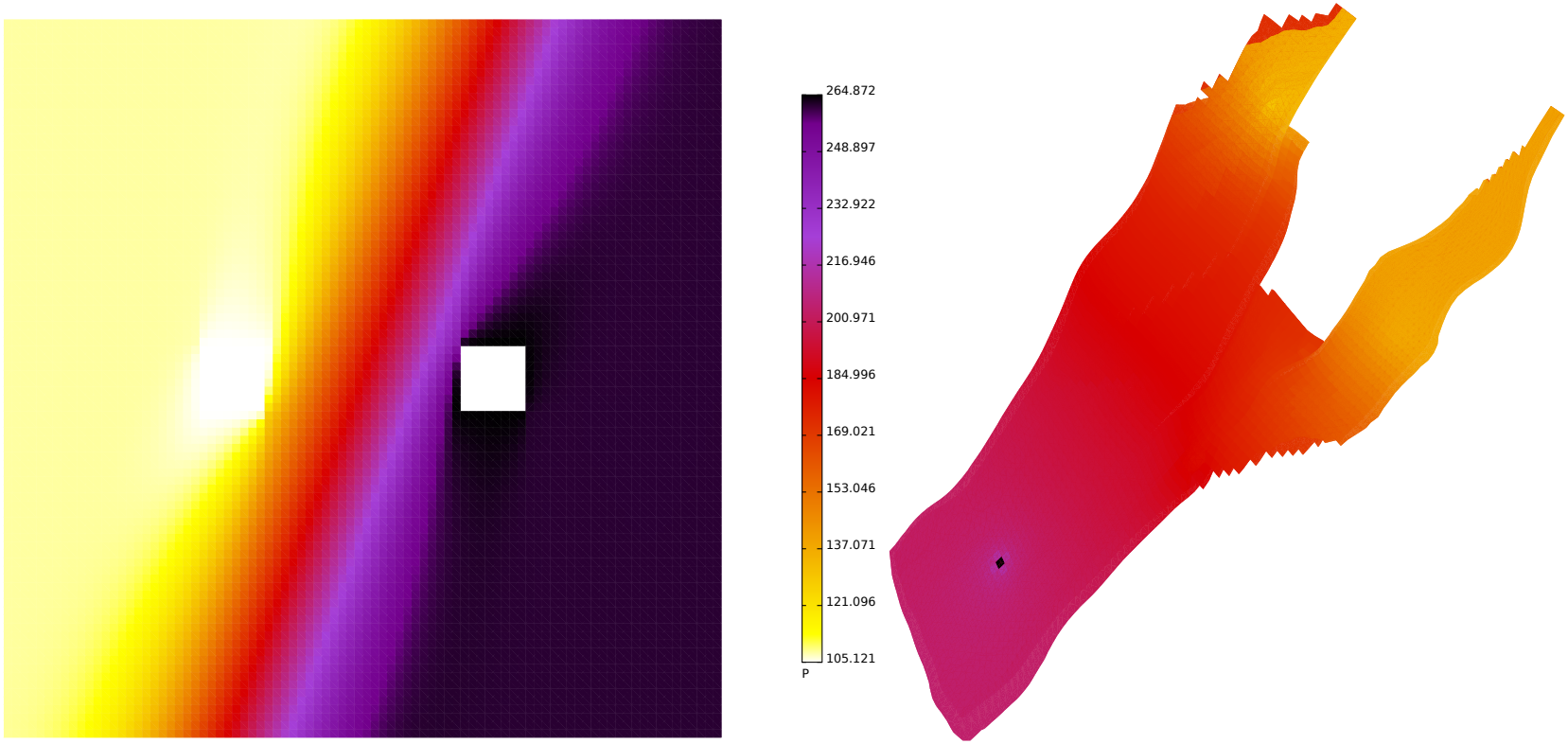

Рис. 2: Задача с двумя скважинами (слева). Задача на данных месторождения Норна (справа).

Примеры генераторов систем

Содержимое ветки lect0, генераторы систем:

poisson.cpp Задача Пуассона на регулярной сетке $N \times N$.

stokes.cpp Задача Стокса на регулярной сетке $N \times M$.

biot.cpp Задача Био на регулярной сетке $N \times M$.

deadoil.cpp Задача двухфазной фильтрации с двумя скважинами на сетке $N \times M$.

biharmonic.cpp Дискретизация бигармонического уравнения на сетке $N \times N$.

Информацию о параметрах можно получить при запуске исполняемого файла без параметров.

Типы уравнений

$$-\Delta p = b \qquad \text{(poisson.cpp)}$$

$$\begin{cases} -\mu\Delta\vec{u} + \nabla p = \mathrm{b} \\ \mathrm{div}(\vec{u}) = 0 \end{cases} \qquad \text{(stokes.cpp)}$$

$$\begin{cases} -\mu\Delta\vec{u} - (\mu+\lambda)\nabla\mathrm{div}(\vec{u}) + \alpha\nabla p = \mathrm{b} \\ \partial_t\left(\zeta p + \alpha\mathrm{div}(\vec{u})\right) - \kappa\Delta p = q \end{cases} \qquad \text{(biot.cpp)}$$

$$\begin{cases} \phi\partial_t S - \mathrm{div}\left(S\kappa\nabla p\right) = q_o \\ \phi\partial_t(1-S) - \mathrm{div}\left((1-S)\kappa\nabla p\right) = q_w \end{cases} \qquad \text{(deadoil.cpp)}$$

$$-\Delta^2 p = b \qquad \text{(biharmonic.cpp)}$$

+ граничные и начальные условия.

Примеры для визуализации решения

Содержимое ветки lect0, создание файла *.vtk*:

scalar_grid.cpp Визуализация решения примеров poisson.cpp и biharmonic.cpp.

stokes_grid.cpp Визуализация решения примеров stokes.cpp и biot.cpp.

deadoil_grid.cpp Визуализация решения примера deadoil.cpp.

Информацию о параметрах можно получить при запуске исполняемого файла без параметров. Для визуализации файлов *.vtk* можно использовать `www.paraview.org`.

Визуализация решения

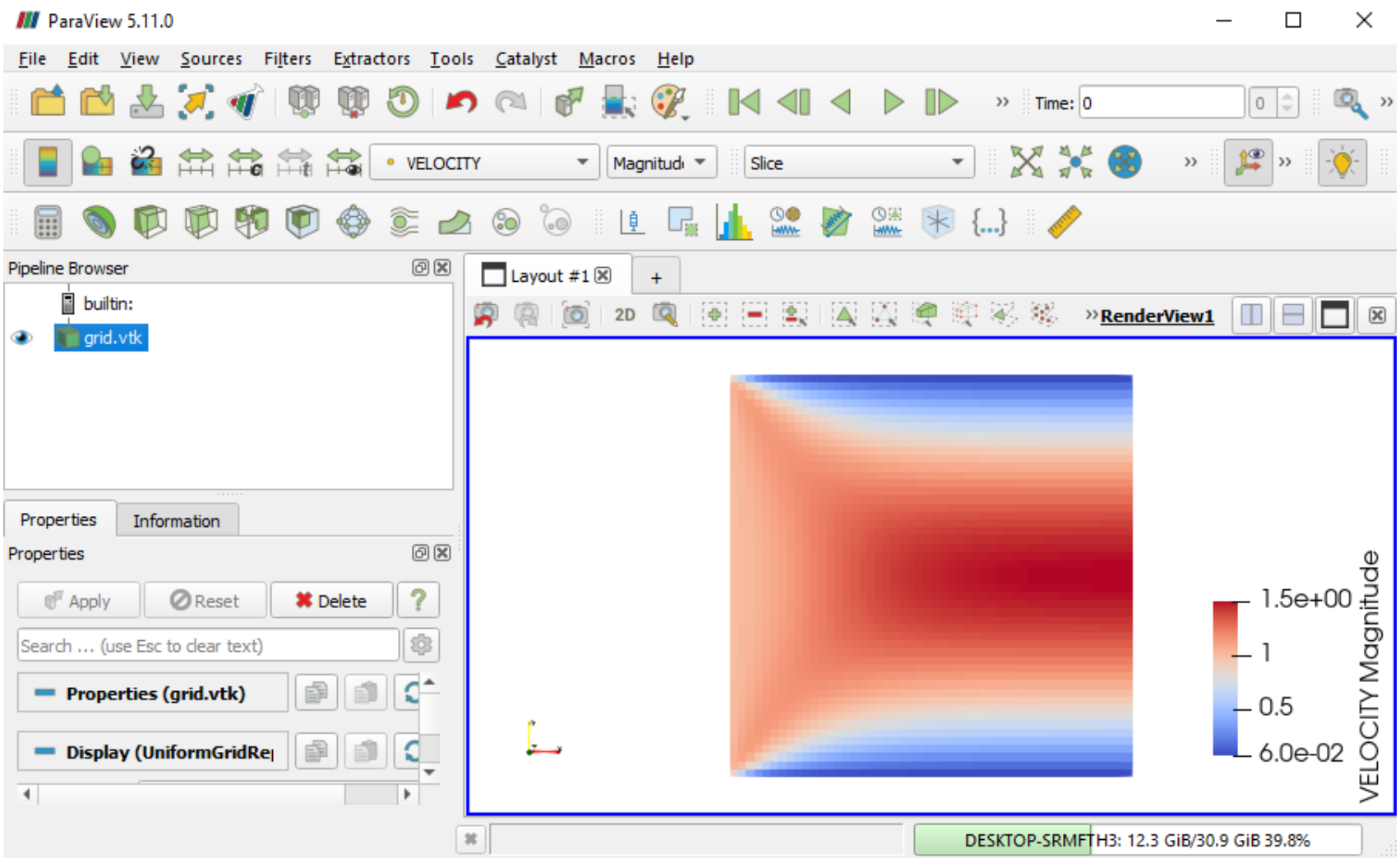

Рис. 3: Пример визуализации решения задачи stokes.cpp с $N = 50$ при помощи программы **paraview**.

Компиляция программ

Необходимые инструменты

- Программа **cmake**: `www.cmake.org`
- Комилятор C++:
  - Linux: g++, clang, intel compiler
  - Windows: microsoft visual studio, clang, windows subsystem for linux, cygwin, intel compiler

Пример компиляции

- git clone `https://github.com/kirill-terekhov/mipt-solvers.git`
- mkdir mipt-solvers-build
- cd mipt-solvers
- git checkout lect0
- cd ../mipt-solvers-build
- **linux** или linux-подобные компиляторы в windows:
  - cmake ../mipt-solvers *-DCMAKE_BUILD_TYPE=Release*
  - cmake –build .
  - Исполняемые файлы появятся в текущей папке.
- **windows** с компилятором из Visual Studio:
  - cmake ../mipt-solvers
  - cmake –build . –config Release
  - Исполняемые файлы появятся в папке Release.

Если создать папку с результатом компиляции внутри папки mipt-solvers, то при переключении на материал другой лекции могут возникнут конфликты в git.

Пример компиляции

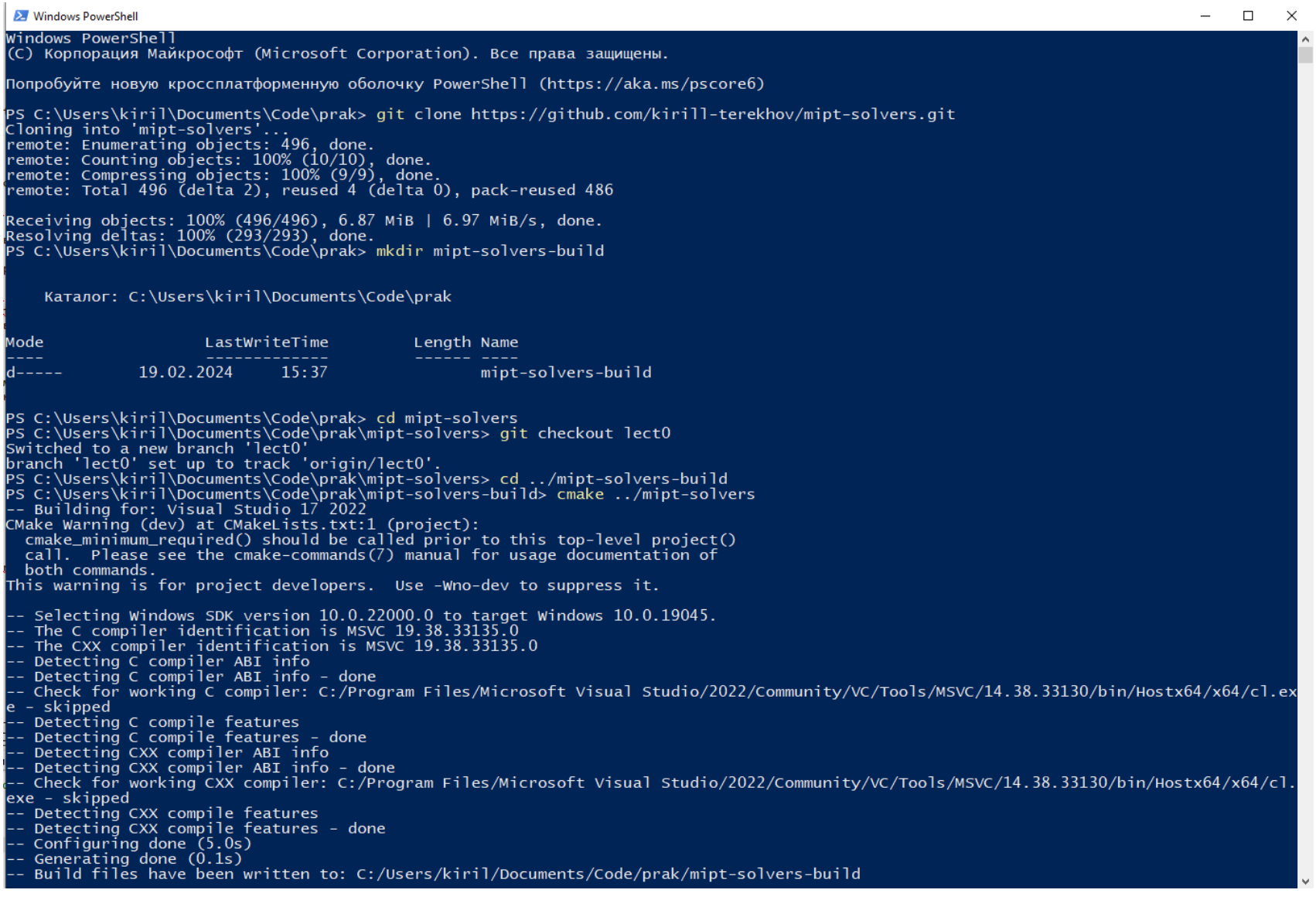

Рис. 4: Пример компиляции на windows.

Пример компиляции

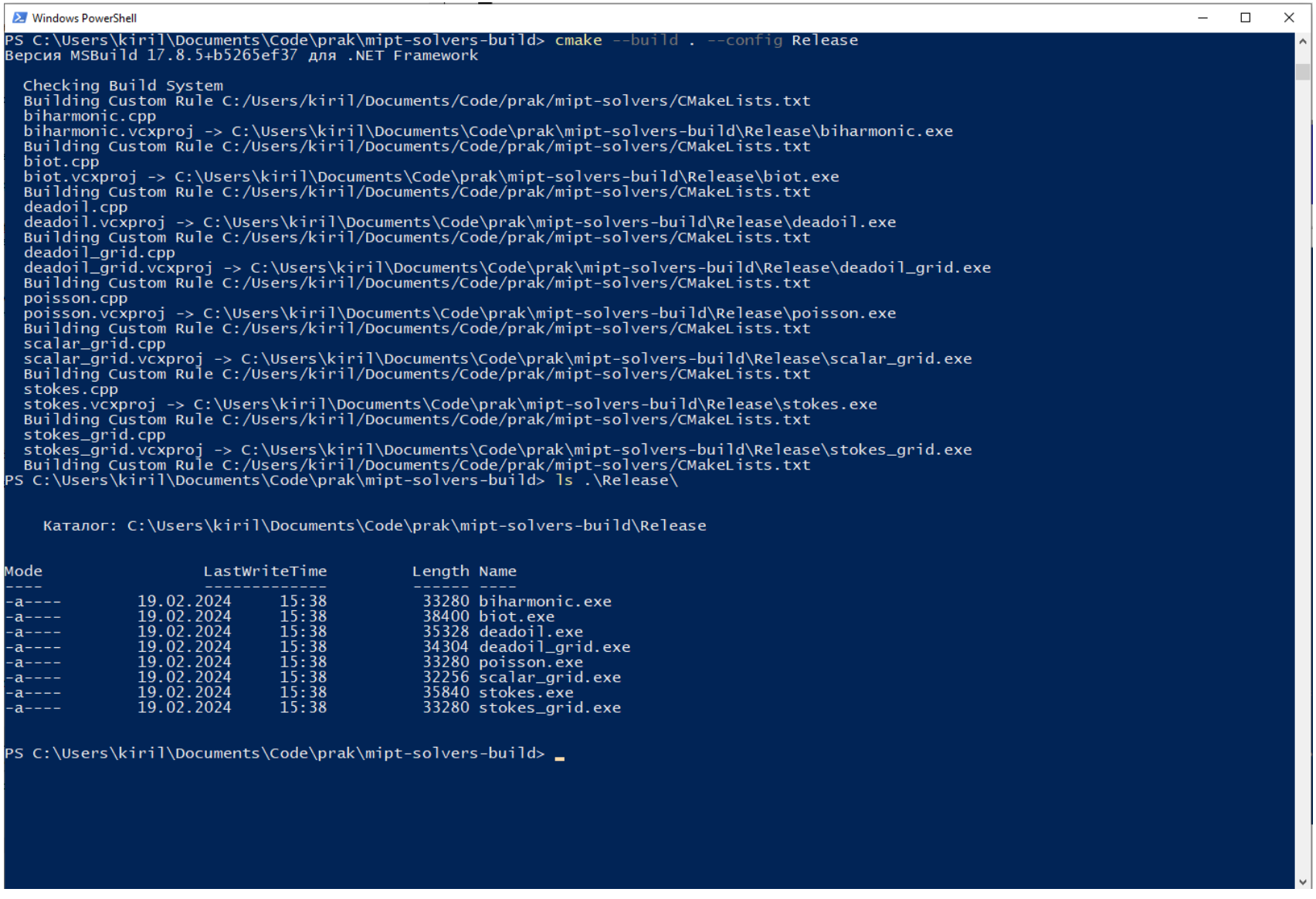

Рис. 5: Пример компиляции на windows.

Пример компиляции

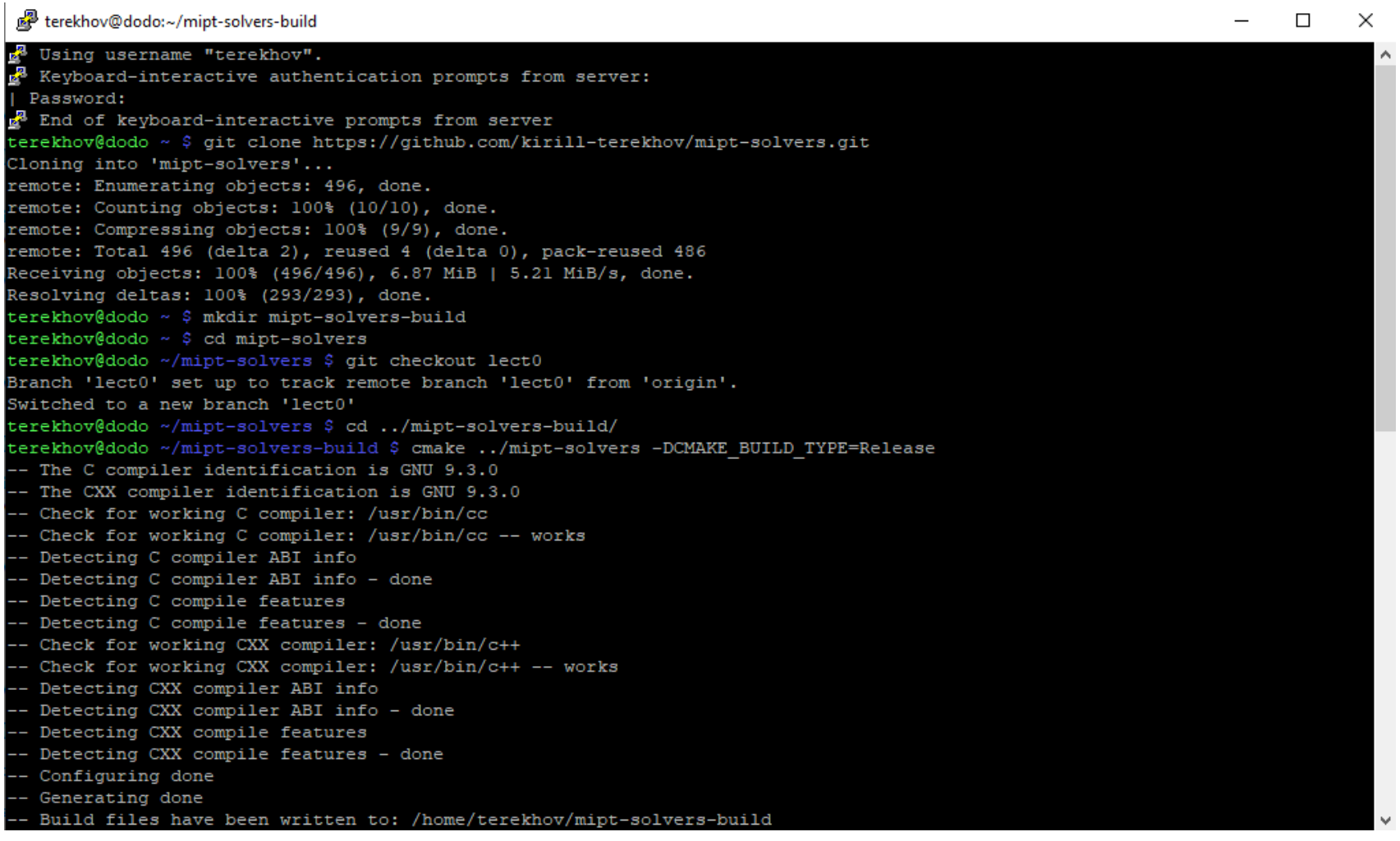

Рис. 6: Пример компиляции на linux.

Пример компиляции

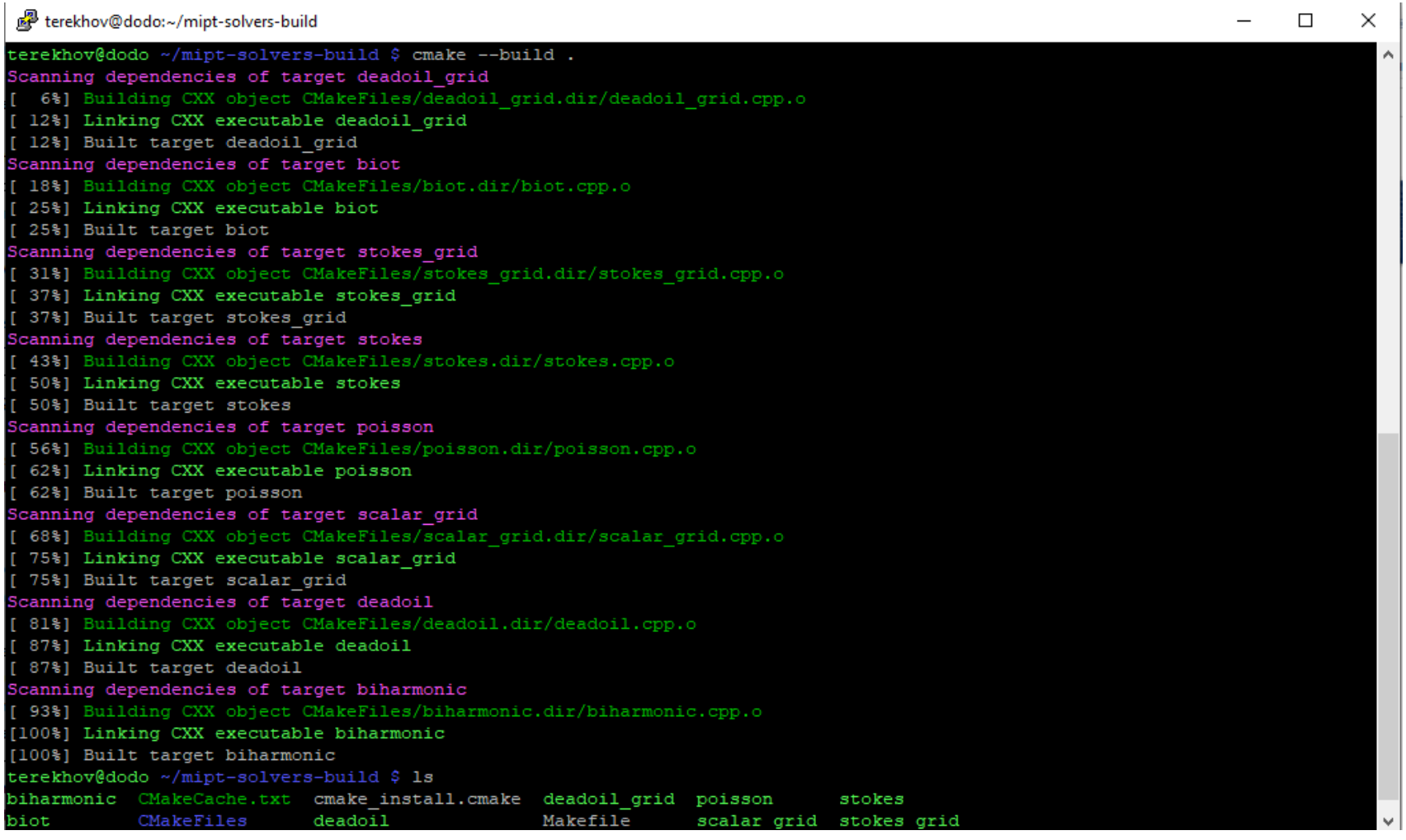

Рис. 7: Пример компиляции на linux.

Переключение материалов лекции

- cd ../mipt-solvers
- git checkout lect1
- cd ../mipt-solvers-build
- cmake –build . *–config Release*

Переключение материалов лекции

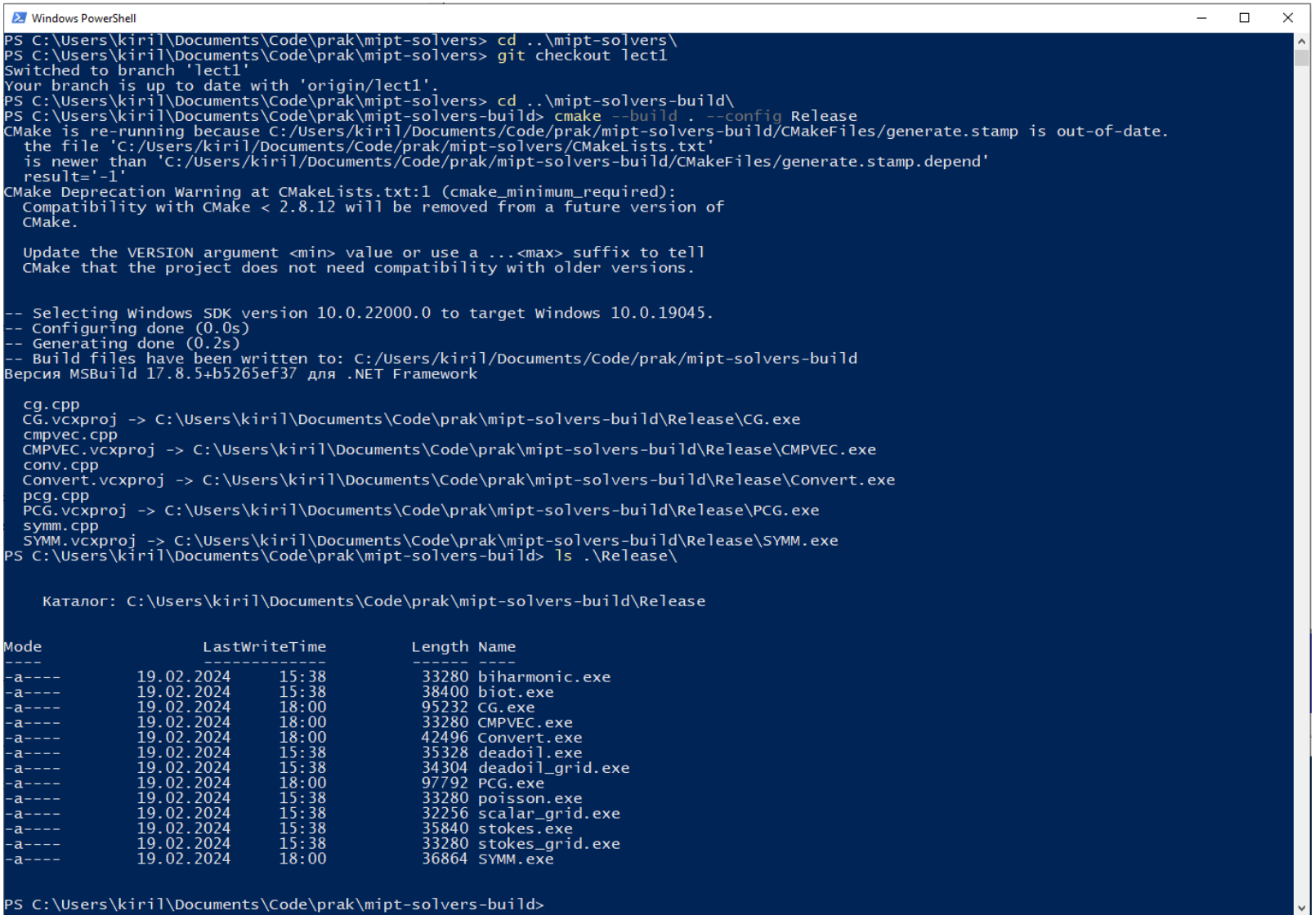

Рис. 8: Пример компиляции на windows.

Переключение материалов лекции

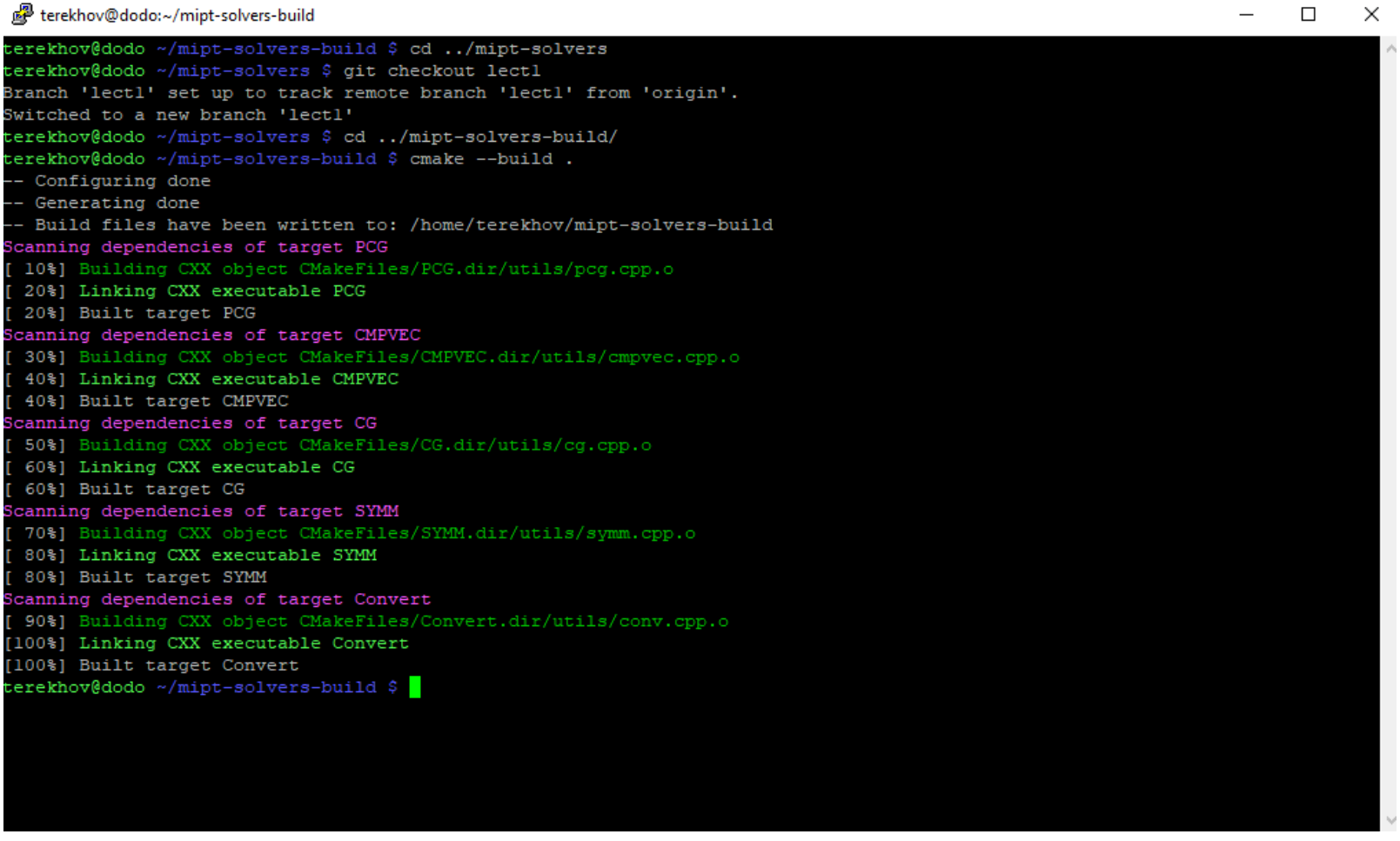

Рис. 9: Пример компиляции на linux.

Разбор заданий

Примеры заданий

Вариант простого задания:

- Статистики по методам в зависимости от роста размера задачи N.
- Из примера biot.cpp сделать симулятор.

Статистика

Используя генераторы систем с параметром N, составить статистику по методам решения:

1. I - количество линейных итераций.
2. T - полное время решение системы.
3. M - занимаемая в процессе решения память.

Запуск теста

- windows:
  - ./Release/poisson.exe 50
  - ./Release/PCG.exe A.mtx b.txt
- linux:
  - ./poisson 50
  - ./PCG A.mtx b.txt

Запуск теста

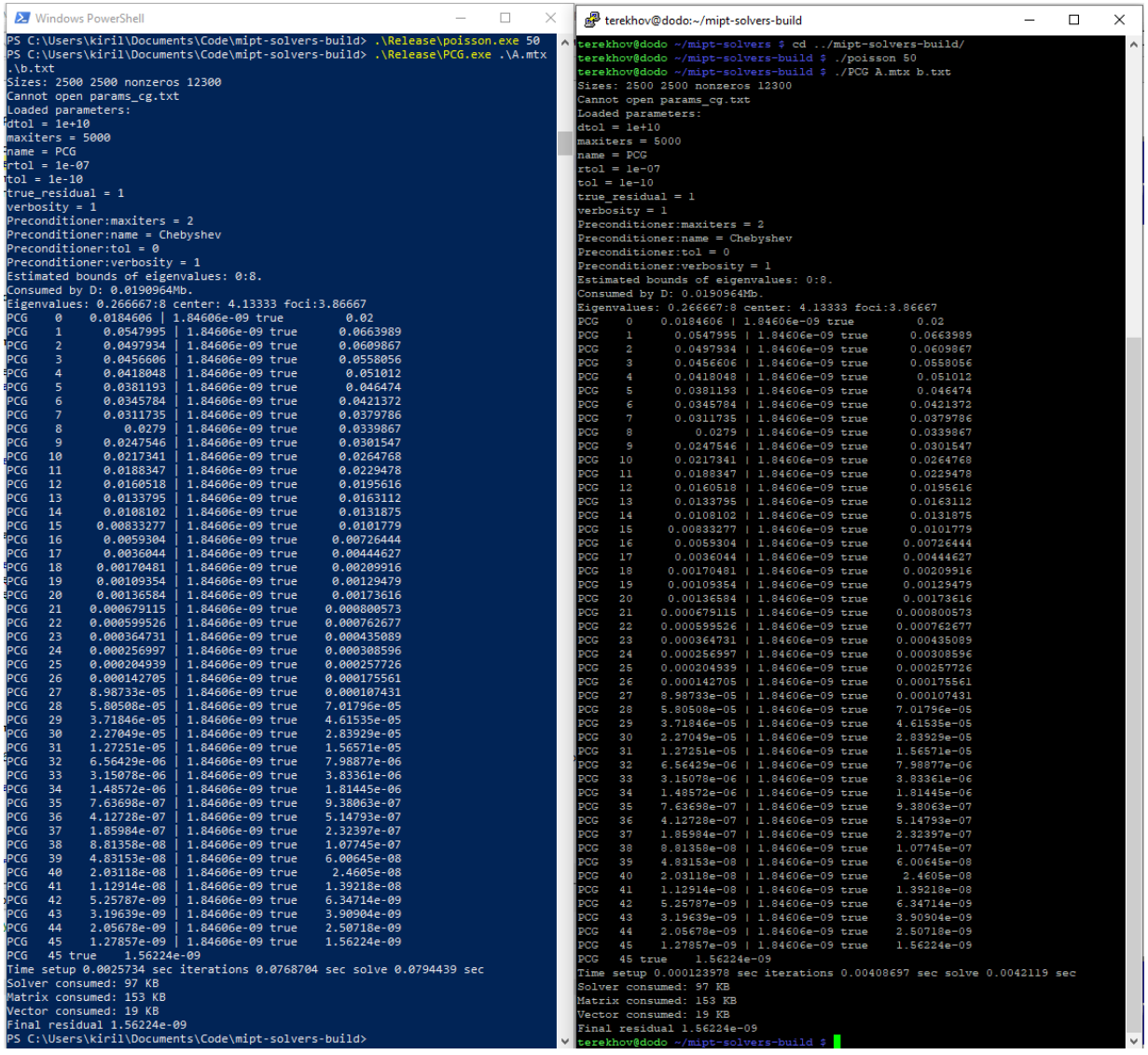

Рис. 10: Запуск примера на windows и linux.

Статистика

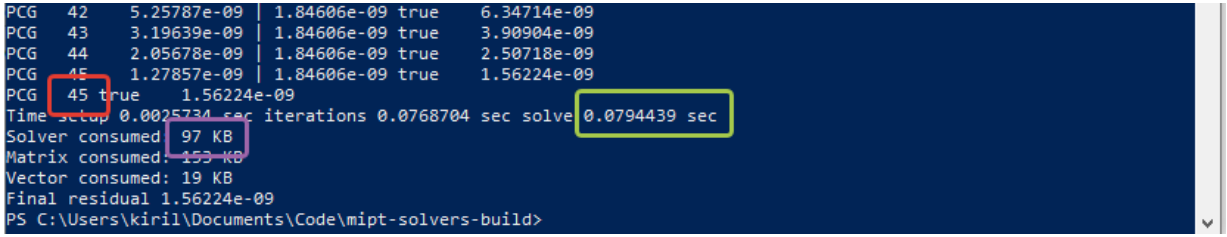

Рис. 11: Данные в примере.

1. I - количество линейных итераций.
2. T - полное время решение системы.
3. M - занимаемая в процессе решения память.

Статистика

N	25	50	100	200	400	800
I	23	45	87	173	347	698
T	$8.25 \cdot 10^{-4}$	$3.72 \cdot 10^{-3}$	$2.54 \cdot 10^{-2}$	0.21	2.12	17.9
M	24	97	390	1562	6250	25000

Таблица 1: Пример таблицы для метода **PCG** и системы **poisson**

Статистика

Изменяя способ и параметр отбрасывания в методах на основе неполной факторизации на фиксированной системе:

1. I - количество линейных итераций.
2. T - полное время решение системы.
3. P - время построения предобуславливателя.
4. M - занимаемая в процессе решения память.

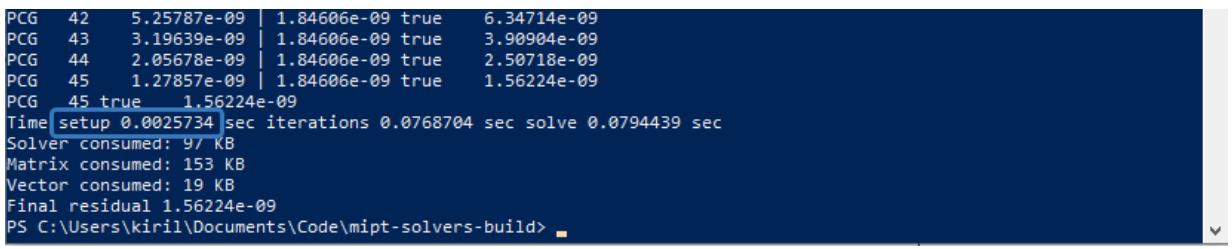

Рис. 12: Данные в примере.

Изменение параметров

Как поменять параметры:

- При запуске исполняемого файла без параметров создается файл **params_default.txt**.
- Исполняемый файл считывает параметры из файла **params.txt**.

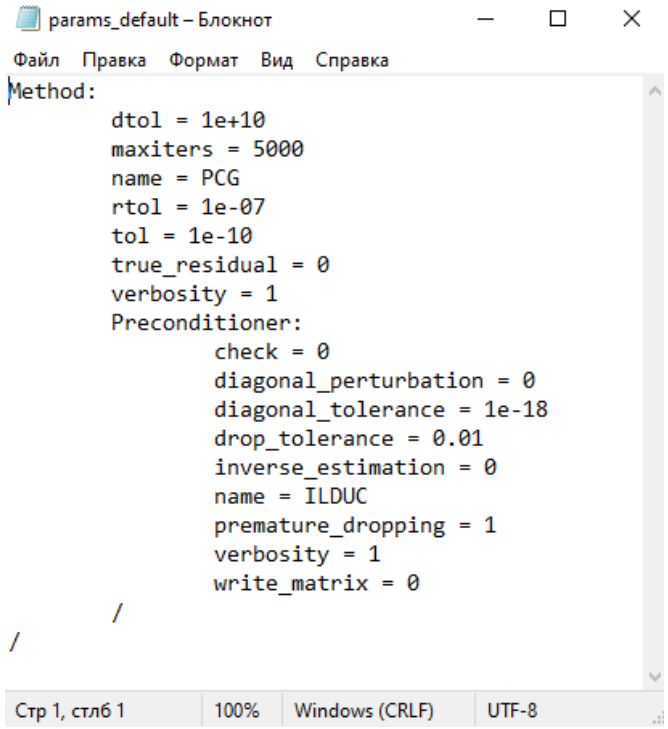

Рис. 13: Параметры **PCG_ILDUC.exe**

Изменение параметров

Как поменять параметры:

- **drop_tolerance** - параметр отбрасывания τ.
- **inverse_estimation** - оценка обратных факторов.

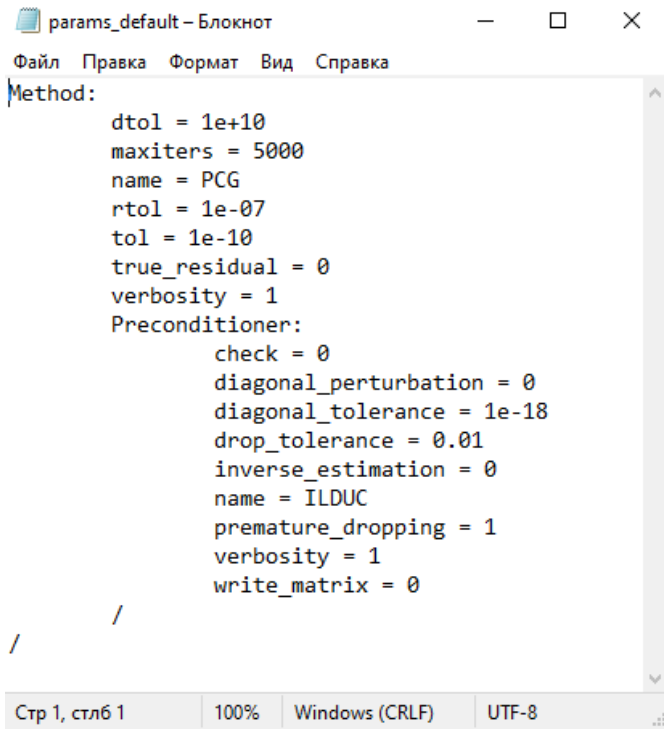

Рис. 14: Параметры **PCG_ILDUC.exe**

Статистика

τ	0.1	0.01	0.001	0.0001	0
I	65	30	15	7	1
T	0.11	$7.5 \cdot 10^{-2}$	0.11	0.23	1.15
P	$3.18 \cdot 10^{-2}$	$4.17 \cdot 10^{-2}$	$8.02 \cdot 10^{-2}$	0.21	1.16
M	1445	2645	5346	11421	25092

Таблица 2: Пример таблицы для метода **PCG_ILDUC** и системы **poisson** с $N = 100$ и **inverse_estimation** $= 0$.

Запуск теста

Для седловых lect10, lect11 и блочных систем lect11

- Для системы Стокса:
  - ./Release/stokes.exe N
  - ./Release/BPCG_AMG_GS.exe $N * (N + 1) * 2$ A.mtx b.txt
- Для системы Био:
  - ./Release/biot.exe N
  - ./Release/FIXED_STRESS_AMG.exe $N * (N + 1) * 2$ A.mtx b.txt
- Для системы двухфазной фильтрации:
  - ./Release/deadoil.exe N
  - ./Release/CPR_*.exe $N * N$ A.mtx b.txt (для всех методов CPR)

Статистика

Выводы по статистике:

1. Какие из методов лучше выбрать для каждой из рассматриваемых систем?
2. Какие из методов применимы к блочным и седловым системам?
3. Какие из методов имеют линейную сложность решения?
4. Как зависит число итераций и время решения от параметра отбрасывания τ в методе неполной факторизации?
5. Результаты в таблицах должны подтверждать выводы.

Выборка тестов

лекция	метод	poisson	biot	stokes	deadoil	biharmonic
lect1	CG	+	+	-	-	+
lect1	PCG	+	+	-	-	+
lect2	PCG_ILDUC	+	±	±	-	±
lect2	BICGSTAB_ILDUC	+	±	±	±	±
lect4	PCG_AMG_CHEB	+	-	-	-	-
lect5	PCG_AMG_JAC	+	-	-	-	-
lect5	PCG_AMG_GS	+	-	-	-	-
lect6	MLILDUCS	+	±	±	±	±
lect7	MLILDUCM	+	±	±	±	±
lect8	MPT_ILDUC	+	±	±	±	±
lect8	SYM_ILDUC	+	±	±	±	±
lect8	MPT_WRCM_ILDUC	+	±	±	±	±
lect8	SYM_WRCM_ILDUC	+	±	±	±	±
lect9	MPT_WRCM_MLILDUCS	+	±	±	±	±
lect9	MPT_WRCM_MLILDUCM	+	±	±	±	±
lect10	BPCG_AMG_GS	-	-	+	-	-
lect10	VANKA	-	-	+	-	-
lect11	CPR_TS_ILU	-	-	-	+	-
lect11	CPR_TS_GS	-	-	-	+	-
lect11	CPR_TSGS_ILU	-	-	-	+	-
lect11	CPR_TSGS_GS	-	-	-	+	-
lect11	FIXED_STRESS_AMG	-	+	-	-	-

Таблица 3: Тесты методов на рост размеры системы.

Варианты с ± могут не решаться или заметно зависеть от параметров и размера системы.

Симулятор

Из примера biot.cpp сделать симулятор.

1. Добавить цикл шагов по времени.
2. Добавить вызов метода для решения системы.
3. Модифицировать правую часть системы в зависимости от решения на прошлом шаге.
4. Вывод файлов сетки в каждый момент времени (как в stokes_grid.cpp).
5. Поэкспериментировать с расстановкой скважин.
6. Пример симулятора двухфазной фильтрации на основе deadoil.cpp находится в ветке lect12. Задача двухфазной фильтрации нелинейная и требует дополнительный цикл для метода Ньютона, в то время как задача Био - линейная.

Симулятор

Пример biot.cpp:

- Система уравнений:

$$-\mu\Delta\vec{u} - (\mu+\lambda)\nabla\mathrm{div}(\vec{u}) + \alpha\nabla p = \mathrm{b},$$
$$\partial_t\left(\zeta p + \alpha\mathrm{div}(\vec{u})\right) - \kappa\Delta p = q,$$

- μ, λ - параметры Ламэ, κ - проницаемость среды, α - коэффициент Био, ζ - ёмкость породы, $\theta = \zeta p + \alpha\mathrm{div}(\vec{u})$ - пористость;
- начальные условия: нулевые, $\mathrm{u}^0 = 0, p^0 = 0$;
- граничные условия на жидкость: непротекание;
- граничные условия на породу: перемещение фиксировано.

Симулятор

Пример biot.cpp:

- Зависимость от прошлого шага по времени в $\partial_t \left(\zeta p + \alpha \mathrm{div}(\vec{u})\right)$.
- Дискретизация по времени:

$$\partial_t \left(\zeta p + \alpha \mathrm{div}(\vec{u})\right) \approx \zeta \frac{p^{n+1} - p^n}{\Delta t} + \alpha \mathrm{div} \left(\frac{\mathrm{u}^{n+1} - \mathrm{u}^n}{\Delta t} \right).$$

- Дискретизация по пространству:

$$\mathrm{div}(\vec{u}) = \partial_x u + \partial_y v \approx \frac{u_{i+1,j} - u_{i,j}}{h} + \frac{v_{i,j+1} - v_{i,j}}{h}$$

- Коэффициенты при неизвестных с p^{n+1} и u^{n+1} попадают в матрицу.
- Прибавка к правой части q:

$$\zeta \frac{p^n}{\Delta t} - \alpha \frac{u^n_{i+1,j} - u^n_{i,j}}{h\Delta t} - \alpha \frac{v^n_{i,j+1} - v^n_{i,j}}{h\Delta t}$$

Симулятор

Пример biot.cpp:

- Формула Писмана для скважины радиуса r_w с забойным давлением p_{bhp}:

$$q = 2\pi\kappa(p - p_{bhp}) / \log\left(\frac{14\sqrt{2}h}{100 r_w}\right).$$

- r_w, κ - вводные параметры, по умолчанию $r_w = 10^{-4}$, $\kappa = 1$.
- Задано положение скважины в середине области $i = N/2, j = M/2$ и давление $p_{bhp} = 100$.
- Можно поменять или добавить еще несколько скважин в других местах и с другим забойным давлением, например, зависящим от времени.

Симулятор

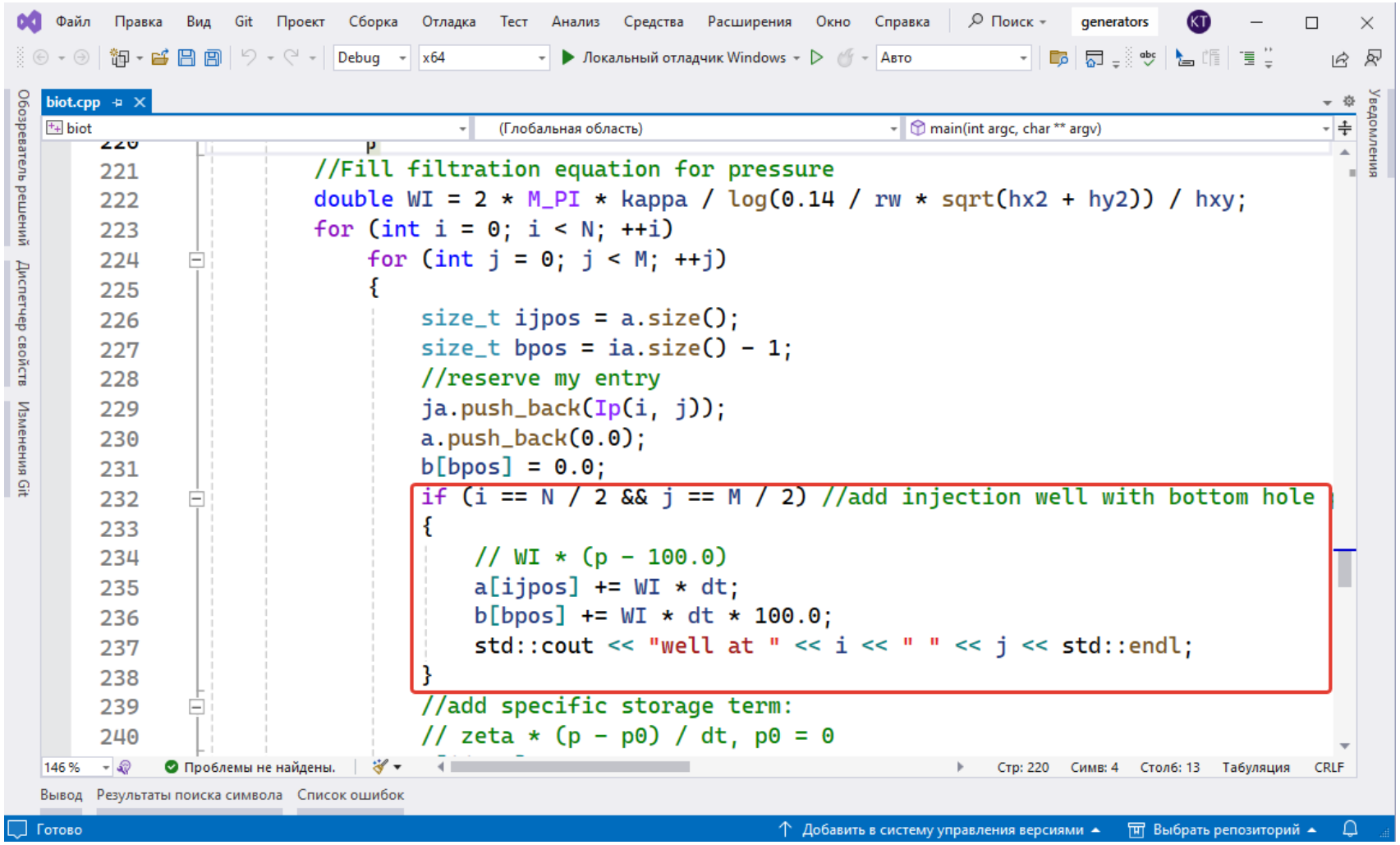

Рис. 15: Участок кода для скважины в biot.cpp.

Симулятор

- В результате работы примера biot.cpp собирается массивы ia, ja, a матрицы в CSR-формате и вектор правой части b.
- Примеры с методами решения принимают на вход объект класса **CSRMatrix**, который имеет конструктор вида **CSRMatrix(ia,ja,a)**.
- В качестве примера передачи матрицы в решатель можно посмотреть на код **utils/deadoil_nln.cpp** в материалах lect12.
- Для визуализации вектора решения можно использовать **stokes_grid.cpp** из материалов lect0.

Python

Пример (Команды в python)

- **scipy.sparse.csr_matrix** – *создать матрицу в формате CSR;*
- **scipy.sparse.coo_matrix** – *создать матрицу в формате COO;*
- **scipy.io.mmread** – *считать файл формата it MatrixMarker;*
- **matplotlib.pylab.spy** – *отрисовать "след" матрицы;*
- **scipy.linalg.solve** – *решить систему (по умолчанию используется пакет UMFPACK).*

Можно перевести формирование системы в python.

Примеры заданий

Вариант сложного задания:

- либо из примера deadoil.cpp сделать симулятор с использованием данных SPE10,
- либо из комбинации примеров biot.cpp и deadoil.cpp сделать новый пример для задачи двухфазной фильтрации в пороупругой среде с использованием данных SPE10.
- Пример симулятора двухфазной фильтрации на основе deadoil.cpp находится в ветке lect12.

Можно перевести формирование системы в python.

Считывание данных SPE10

Данные SPE10:

- Ссылка на данные `https://www.spe.org/web/csp/datasets/set02.htm#download`.
- В архиве лежат файлы **spe_phi.dat** и **spe_perm.dat** в ASCII формате.
- Файл **spe_phi.dat** содержит информацию о пористости θ в каждой ячейке.
- Файл **spe_perm.dat** содержит информацию о тензоре $\boldsymbol{\kappa}$ в каждой ячейке.

Считывание данных SPE10

Данные SPE10:

- Данные проницаемости заданы в виде диагональной матрицы:

$$\boldsymbol{\kappa} = \begin{bmatrix} \kappa_x & & \\ & \kappa_y & \\ & & \kappa_z \end{bmatrix}, \quad \kappa_x = \kappa_y \neq \kappa_z.$$

- Будем считать, что проницаемость скалярная, возьмем $\kappa = \kappa_x$.
- В данных есть непроницаемые ячейки с нулевыми θ и κ: можно заменить их минимальными значениями, например, 10^{-4}, чтобы система была не сингулярна,
- Размеры массивов: $60 \times 220 \times 85 = 1\,122\,000$ ячеек, можно взять только один слой, например 45-ый.

Считывание данных SPE10

Данные SPE10:

- Размеры массивов: $60 \times 220 \times 85 = 1\,122\,000$ ячеек, можно взять только один слой данных по вертикали, например, 45-ый.
- Для уменьшения задачи из слоя можно выделить отдельный участок данных.
- Физические размеры ячеек: $20ft \times 10ft \times 2ft$.

Считывание данных SPE10

Пример (Считывание пористости в C++)

- *std::ifstream fporo("spe_phi.dat");*
- *double poro[60][220][85];*
- *for (int k = 0; k < 85; ++k)*
- *for (int j = 0; j < 220; ++j)*
- *for (int i = 0; i < 60; ++i) fporo » poro[i][j][k];*
- *fporo.close();*

Считывание данных SPE10

Пример (Считывание проницаемости в C++)

- *std::ifstream fperm("spe_phi.dat");*
- *double perm[60][220][85][3];*
- *for (int l = 0; l < 3; ++l)*
- *for (int k = 0; k < 85; ++k)*
- *for (int j = 0; j < 220; ++j)*
- *for (int i = 0; i < 60; ++i) fperm » perm[i][j][k][l];*
- *fperm.close();*

Считывание данных SPE10

Пример (Считывание в python)

- *import numpy as np*
- *phi = np.loadtxt("spe_phi.dat").reshape((60 * 220 * 85)).reshape((60,220,85),order='F')*
- *perm = np.loadtxt("spe_perm.dat").reshape((60 * 220 * 85 * 3)).reshape((60,220,85,3),order='F')*

Считывание данных SPE10

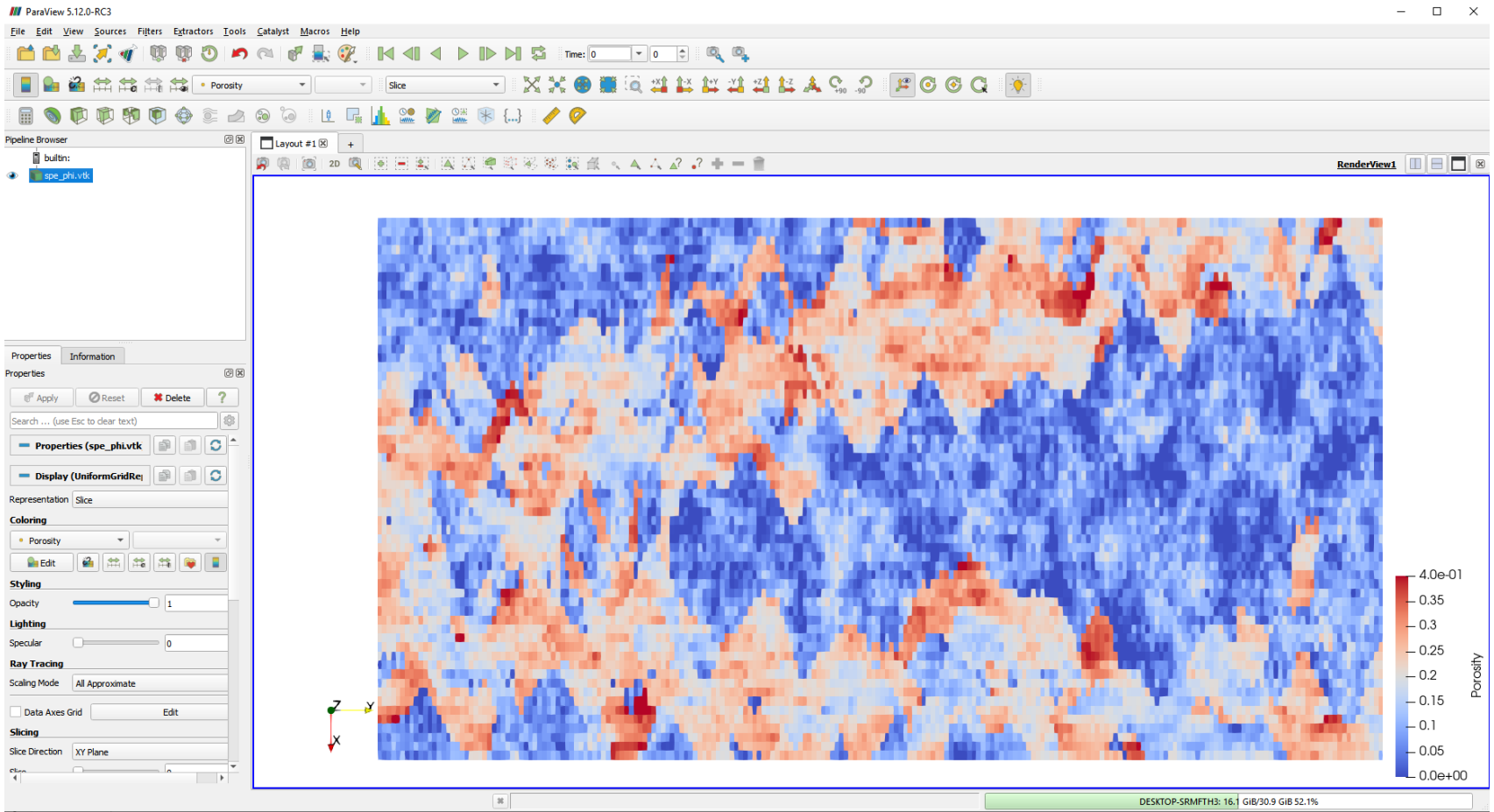

Рис. 16: 45-ый слой данных пористости из данных SPE10 в paraview.

Симулятор

Из примера deadoil.cpp сделать симулятор.

1. Учесть неоднородность данных пористости и проницаемости.
2. Добавить цикл шагов по времени.
3. Добавить цикл метода Ньютона решения нелинейной задачи.
4. Добавить вызов метода для решения системы.
5. Модифицировать правую часть системы в зависимости от решения на прошлом шаге.
6. Расположить скважины.
7. Пример двухфазной фильтрации на основе deadoil.cpp находится в ветке lect12: можно сделать на основе этого примера.

Уравнение двухфазной фильтрации

Примеры deadoil.cpp и lect12:

- Описывают вытеснение нефти водой.
- Неизвестные: $S \in [0,1]$ - насыщенность нефти, p - давление воды.
- Система уравнений:

$$\begin{cases} \partial_t \theta S - \operatorname{div}(S\kappa\nabla p) = q_o \\ \partial_t \theta (1-S) - \operatorname{div}((1-S)\kappa\nabla p) = q_w \end{cases}$$

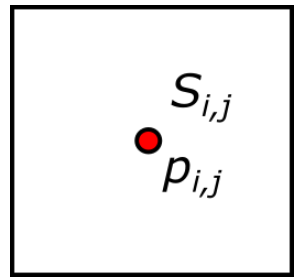

Рис. 17: Степени свободы

- Граничные условия: непротекание.
- Скважина радиуса r_w с забойным давлением p_{bhp}: $q = 2\pi\kappa(p - p_{bhp})/\log\left(\frac{14\sqrt{2}h}{100 r_w}\right)$, $q_o = Sq$, $q_w = (1-S)q$.
- θ, κ - данные из SPE10.

Численная схема

Примеры deadoil.cpp и lect12:

- Определим поток: $q = -\kappa \mathbf{n} \cdot \nabla p$.
- На регулярной прямоугольной сетке с шагом h и неоднородным коэффициентом κ:

$$q = -\kappa \mathbf{n} \cdot \nabla p = -\kappa \frac{\partial p}{\partial x}.$$

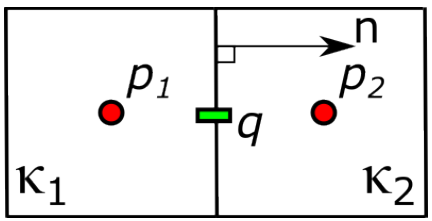

Рис. 18: Поток

- Введем давление на грани p_f и шаг сетки h, тогда:

$$q \approx \kappa_1 \frac{p_1 - p_f}{h/2} = \kappa_2 \frac{p_f - p_2}{h/2}.$$

- Решим относительно p_f и подставим в q:

$$p_f = \frac{\kappa_1 p_1 + \kappa_2 p_2}{\kappa_1 + \kappa_2}, \quad q = \frac{2\kappa_1 \kappa_2}{\kappa_1 + \kappa_2} \frac{p_1 - p_2}{h}.$$

Уравнение Био

Связка deadoil.cpp и biot.cpp:

- Неизвестные: смещение породы $\vec{u} = [u, v]^T$ и давление жидкости p.
- Система уравнений:

$$\begin{cases} -\mu\Delta\vec{u} - (\mu + \lambda)\nabla\mathrm{div}(\vec{u}) + \alpha\nabla p = \mathrm{b} \\ \partial_t\theta S - \mathrm{div}\left(S\kappa\nabla p\right) = q_o \\ \partial_t\theta(1 - S) - \mathrm{div}\left((1 - S)\kappa\nabla p\right) = q_w \end{cases}$$

- $\theta = \theta_0 + \zeta p + \alpha\mathrm{div}(\vec{u})$ - пористость.
- θ_0, κ - данные из SPE10.
- μ, λ - заданные константы.

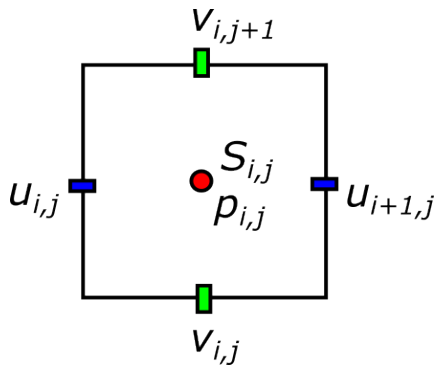

Рис. 19: Степени свободы