Практическое задание 4

Рыбка Елизавета, 474

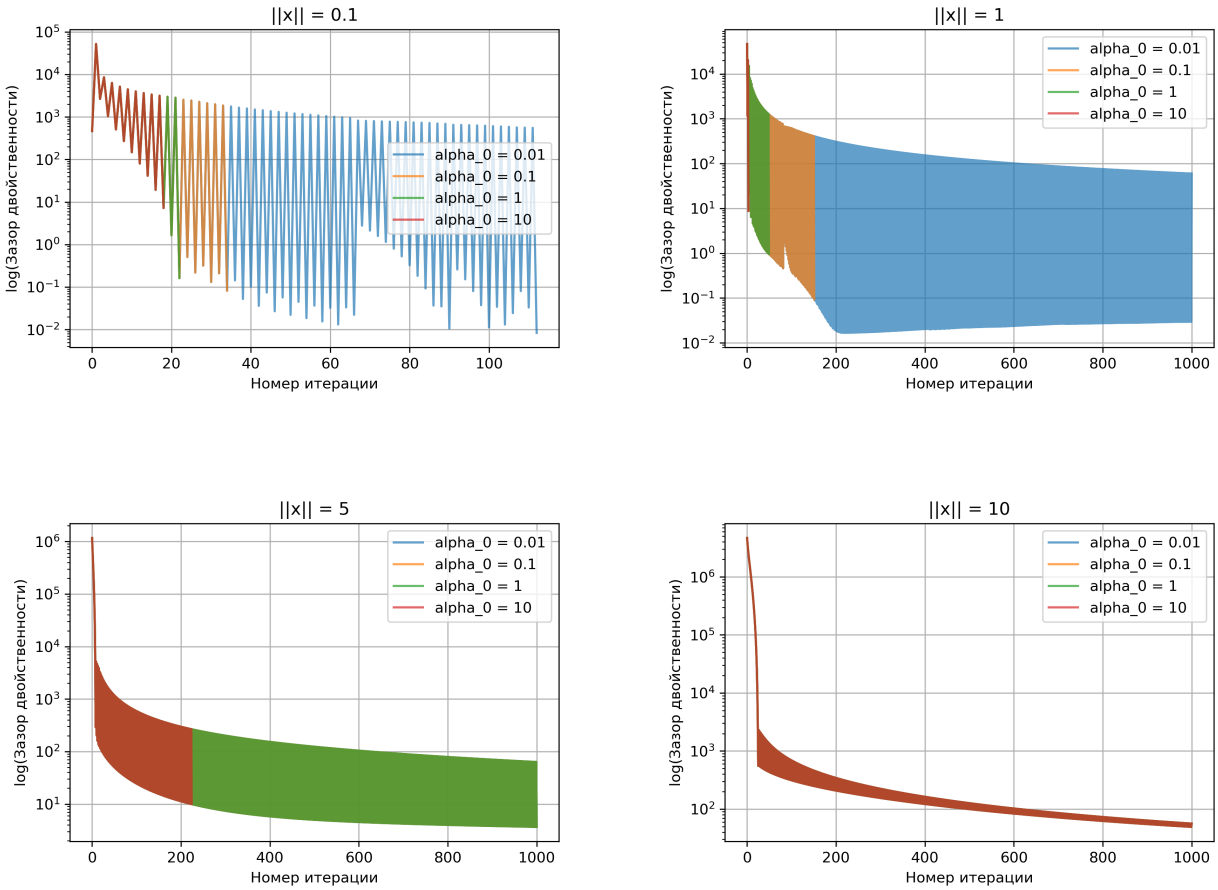
19 декабря 2017 г.

Эксперимент 1: Выбор длины шага в субградиентном методе

Генерируем данные:

- Матрица A размера $m \times n$ с помощью функции $np.random.rand$ т.е. каждый элемент матрицы случайным образом выбран из нормального распределения на $[0, 1)$.
  $n = 1000, m = 100$
- b — нулевой вектор $\in \mathbb{R}^m$
- $x_0 \in \mathbb{R}^n$ также создается с помощью $np.random.rand$. Затем производится нормировка $x_0 = d\frac{x_0}{\|x_0\|}$. Т.е. в итоге получается вектор сонаправленный исходно сгенерированному, но длины d

Приведем графики зависимости работы субградиентного метода в зависимости от константы α_0 для различных начальных точек. Т.к b — нулевой вектор x^* находится около нуля. Поэтому $\|x_0\|$ можно считать расстоянием от начальной точки до искомой. $\lambda = 0.5$



Выводы:

Построенные графики полупрозрачны, поэтому из однородности последнего($\|x_0\| = 10$) делаем вывод, что метод работает одинаково для всех протестированных α_0 при данном $x_0, \|x_0\|$. В целом из графиков можно сделать вывод, что лучше работает большое α_0, но до тех пор пока оно больше $\|x_0\|$. При пересечении же этой границы разницы нет. ~~Почему??~~ Т.к. в реальной жизни мы не знаем насколько далеко наша начальная точка находится от искомой, то по большому счету этот анализ нам ничего не дает.

Эксперимент 2: Среднее число итераций линейного поиска в схеме Нестерова

Генерируем данные:

- Матрица A размера $m \times n$ с помощью функции *np.random.rand* т.е. каждый элемент матрицы случайным образом выбран из нормального распределения на $[0, 1)$. $n = 1000, m = 1000$
- $b \in \mathbb{R}^m$ создается с помощью *np.random.rand*
- $x_0 \in \mathbb{R}^n$ — нулевой вектор

Представим результаты на графиках:

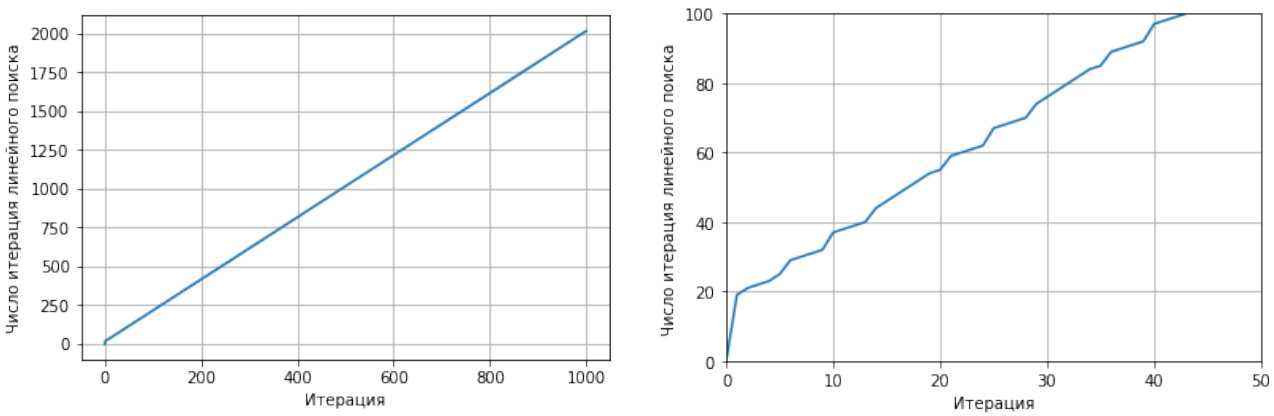


Рис. 1: Зависимость суммарного числа итераций линейного поиска от числа итераций простого метода. Полный график (слева) и приближенный график (справа)

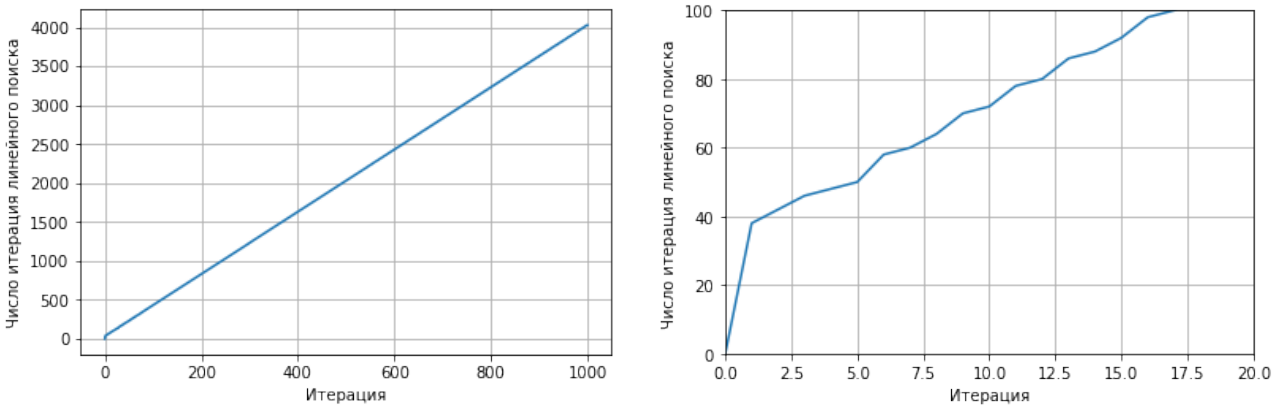


Рис. 2: Зависимость суммарного числа итераций линейного поиска от числа итераций ускоренного метода. Полный график (слева) и приближенный график (справа)

Среднее число итераций для простого метода: 2.012
Среднее число итераций для ускоренного метода: 4.026

Выводы: Видим, что среднее число итераций линейного поиска действительно примерно равно 2. В случае же ускоренного метода получается примерно 4[1].

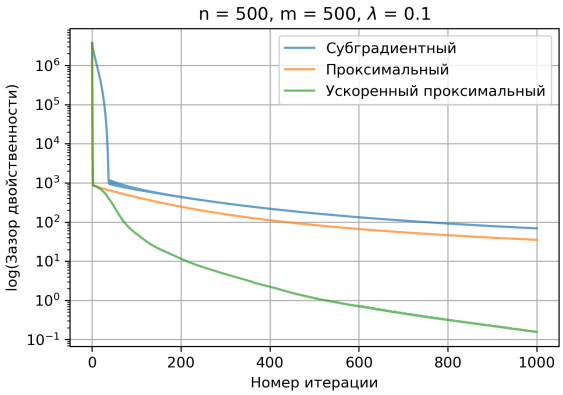
Эксперимент 3: Сравнение методов

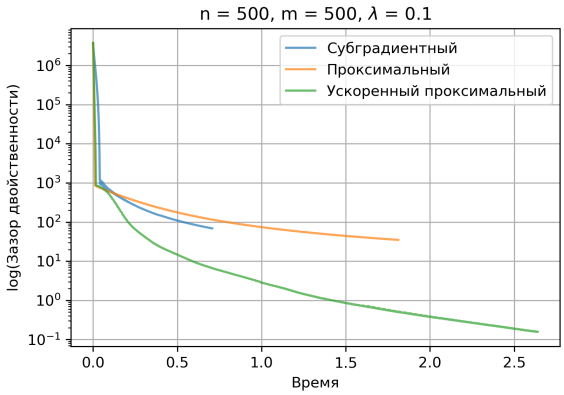
Генерируем данные:

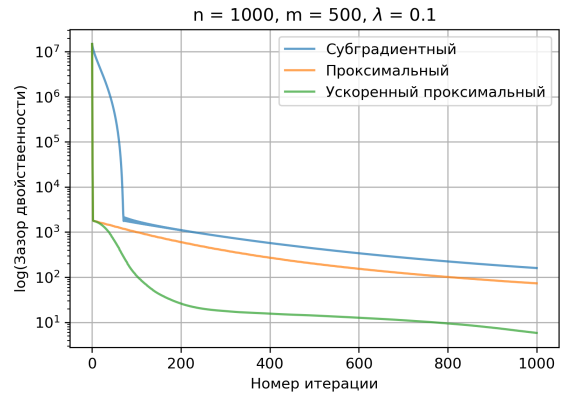
- Матрица A размера $m \times n$ с помощью функции $np.random.rand$ т.е. каждый элемент матрицы случайным образом выбран из нормального распределения на $[0, 1)$.
- b — нулевой вектор $\in \mathbb{R}^m$
- $x_0 \in \mathbb{R}^n$ также создается с помощью $np.random.rand$.

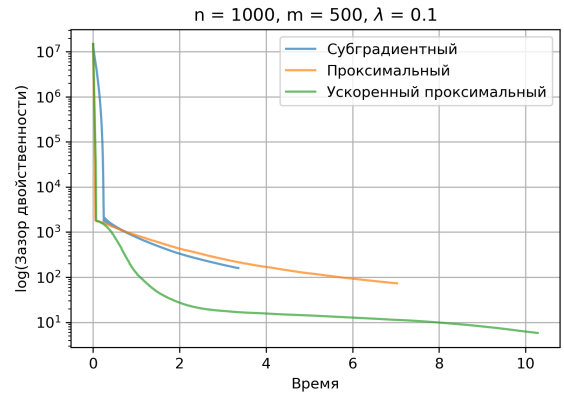
[1] ???

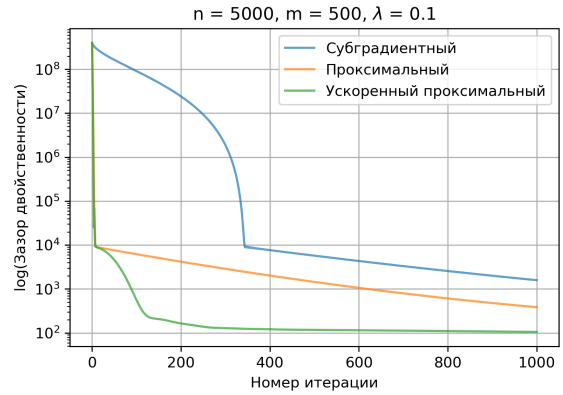
Зависимость от n

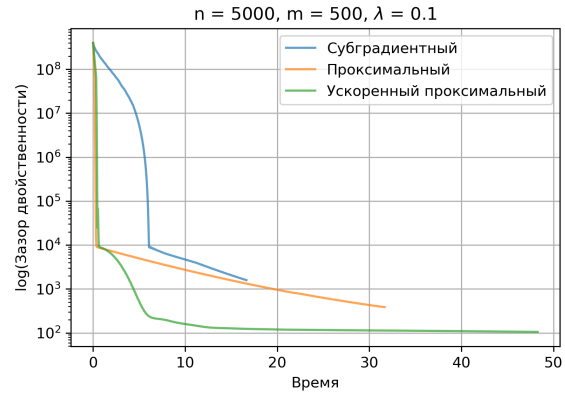








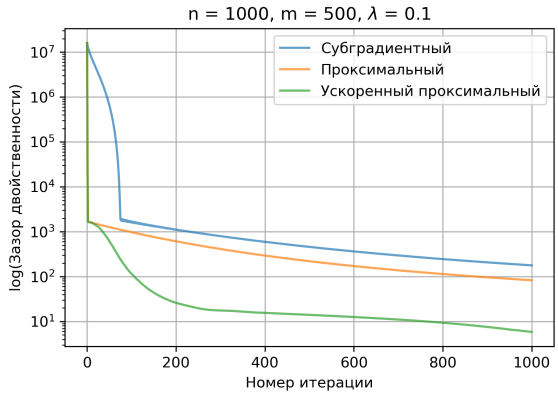


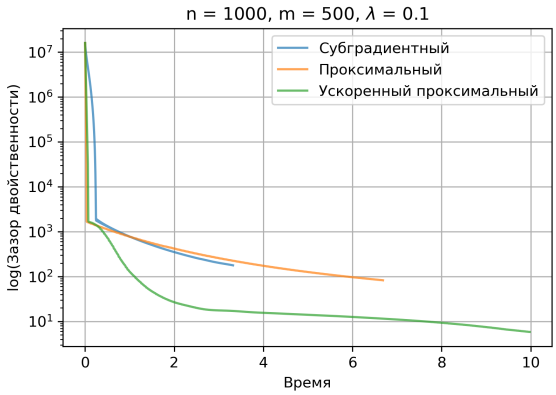


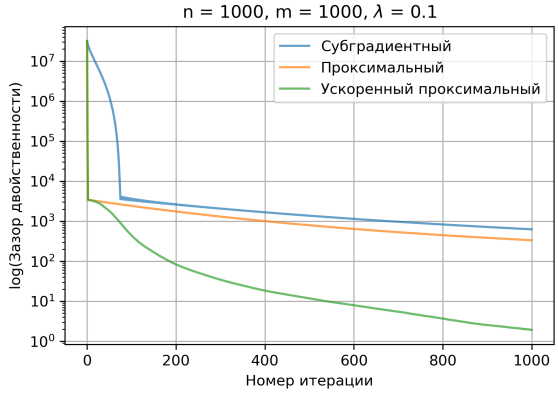
Выводы:

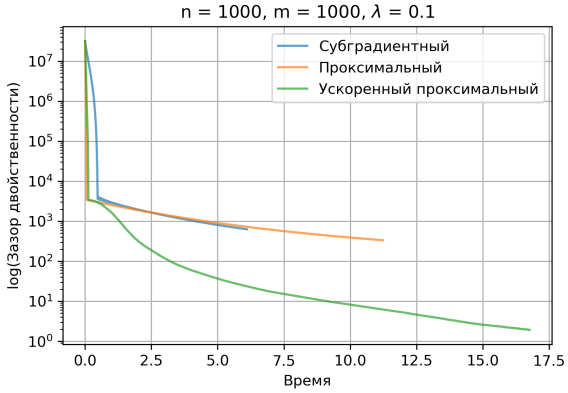
Видим, что ускоренный проксимальный метод работает лучше всего при любом n. По времени два других метода заканчивают раньше (это всего лишь показывает что одна итерация выполняется быстрее), но достигают гораздо меньшей точности. За одно и то же время ускоренный проксимальный метод может больше.Увеличение n влияет на его скорость сходимости меньше чем на остальные методы.

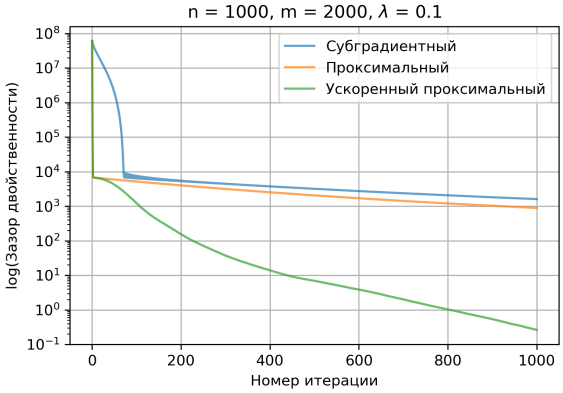
Зависимость от m

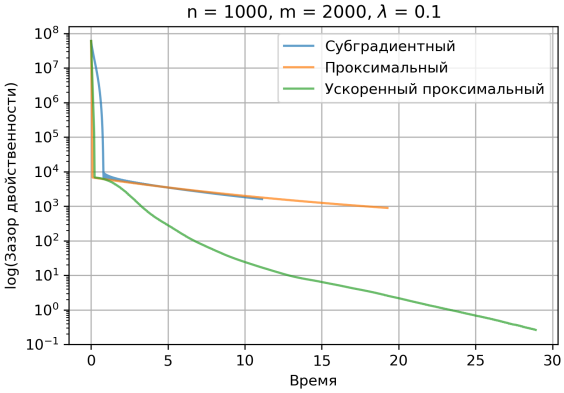










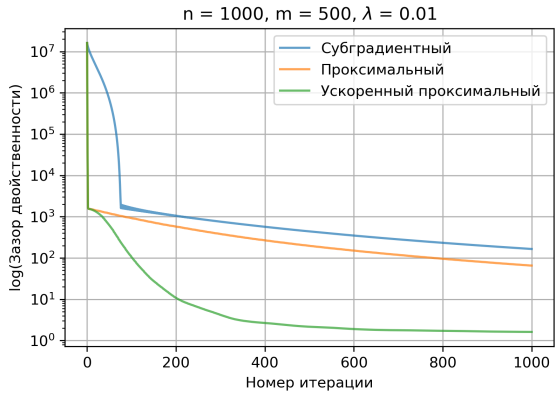


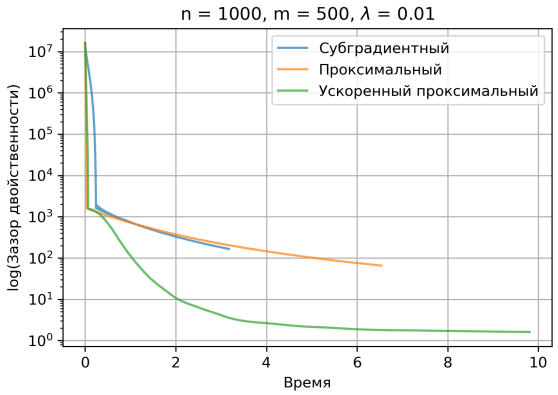
Выводы:

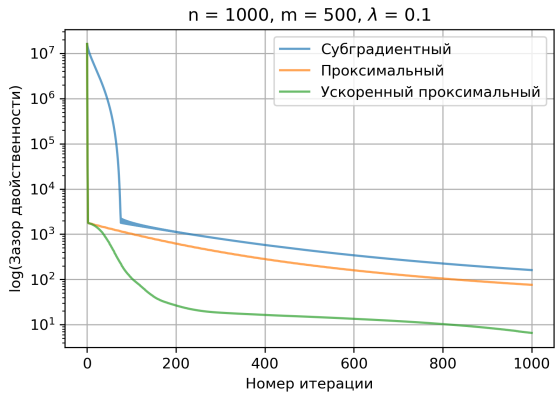
Для ускоренного проксимального метода видим зависимость от m аналогичную зависимости от n. Это объясняется тем, что сложность по m, n определяется сложностью матричновекторного умножения, т.е. $O(mn)$

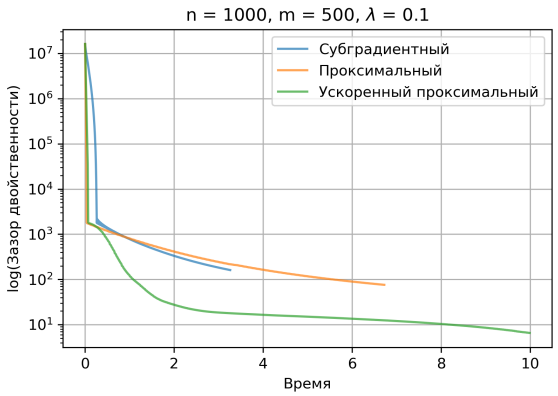
На проксимальный градиентный метод зависит от m сильнее чем от n. При увеличении m за одно и то же количество итераций он достигает меньшей точности.

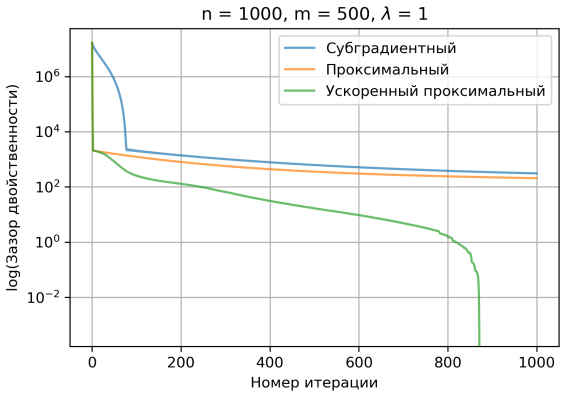
Зависимость от λ

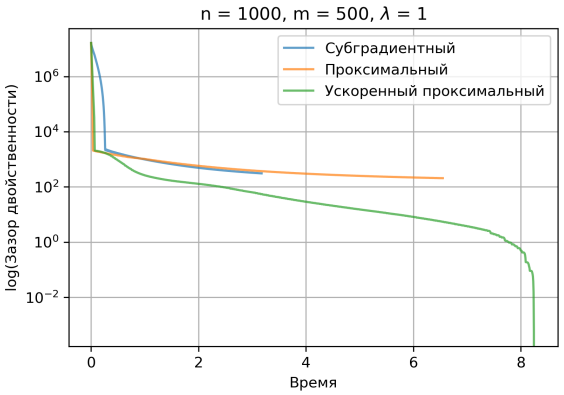


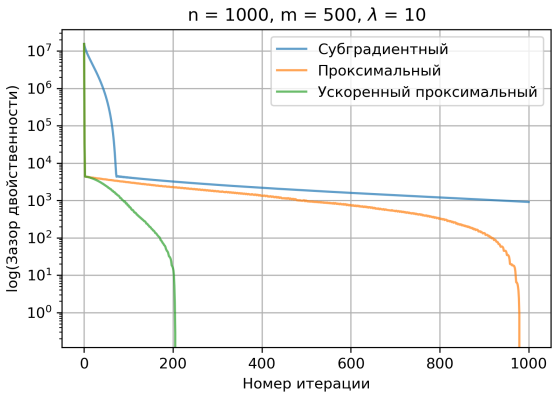


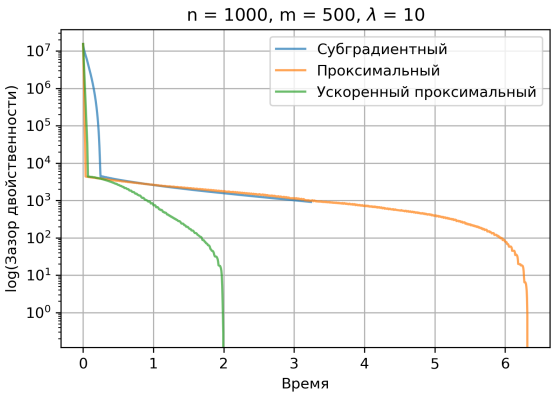


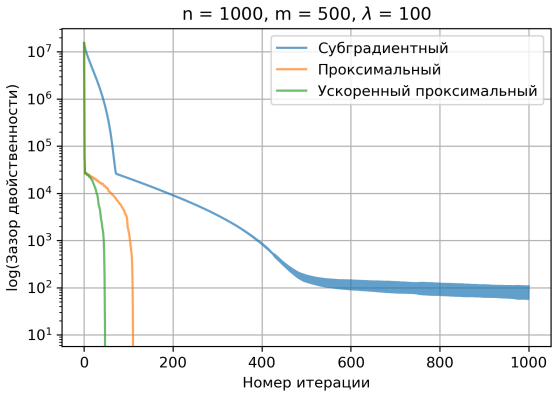


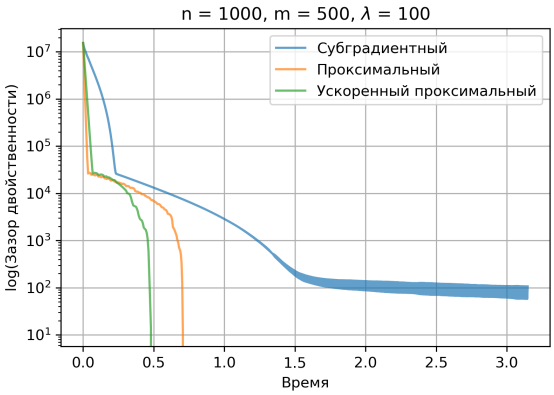












Выводы:

Увеличение λ ведет к отбору признаков (занулению части координат). Если λ то многие признаки быстро становятся нулями, а т.к. наш минимум в центре координат, мы получаем резкий спуск метода.