

При прогонке для матриц размером от 1000×1000 до 1100×1100 с шагом единица для стандартного и блочного методов перемножения наблюдается всплеск времени выполнения при размере матрицы 1024×1024 (вероятно critical size). Этот же всплеск имеет место при замене float на double. В обоих случаях для стандартного метода присутствуют еще несколько скачков, которые по картинке кажутся закономерными, однако после проверки цифр, кажутся скорее бессистемными. Интересным наблюдением является то, что при использовании float блочный метод перемножения оказывается менее эффективным, однако не подвержен скачкам (исключая матрицы размером 1024×1024). Если отбросить скачки, то для всех трех методов время выполнения растет скорее линейно.

При прогонке для матриц размером от 1000×1000 до 2000×2000 с шагом десять никаких особых аномалий не наблюдается. Однако, всплески и падения в скорости выполнения при замене float на double для стандартного метода становятся более системными. В частности при использовании double алгоритм заметно ускоряется с шагом 40. На данных размерах матриц блочный метод всегда оказывается более эффективным, чем стандартный, хотя при использовании float есть несколько размерностей, на которых стандартный метод перемножения выходит вперед. То же самое мы наблюдали и при прогонке от 1000×1000 до 1100×1100 с шагом единица. Однако блочный метод не подвержен скачкам скорости выполнения. Хотя исключением все же является размерность около 1260×1260 для которой во всех случаях наблюдалось некоторое замедление, включая swar метод. Похожее поведение также наблюдалось около размерности 1900×1900 . В обоих случаях блочный и swar метод испытывали куда меньшее замедление, чем стандартный, а при смене float на double для swar метода замедление и вовсе исчезло.

Для всех случаев наибольшую эффективность показал swar метод перемножения. Он всегда оказывался быстрее стандартного и блочного методов выполнения, практически не подвержен скачкам скорости и его время выполнения растет линейно.