

Применение современных методов программирования в вычислительном комплексе NOISETTE

Краснов Михаил Михайлович, ИПМ им. М.В. Келдыша РАН

Программный комплекс NOISETTE предназначен для крупномасштабных расчётов задач аэродинамики и аэроакустики. Этот комплекс позволяет решать очень широкий класс задач, причём после некоторой модификации комплекса с его помощью можно решать задачи, выходящие за первоначально заданные рамки. В частности, в последнее время были успешно решены задача теплопроводности в твёрдом теле и задача течения многокомпонентного газа.

Тем не менее, первоначальная архитектура системы была такова, что добавление новых реализаций того или иного типа объектов (решателей, источников, вязких членов и т.д.) требовало внесения изменений в основной исходный код, причём в нескольких местах. Это приводило к тому, что разработка своих реализаций объектов сторонними коллективами было затруднительно. После внесения нужных изменений в основной код сложно было брать новые версии основного кода у разработчиков комплекса, т.к. после этого требовалось повторное внесение нужных изменений в основной код. При разработке нашим коллективом многокомпонентного течения разработчики основного кода пошли нам навстречу и вставили наш код в основной код комплекса. А при решении задачи теплопроводности в твёрдом теле нужные исправления были внесены непосредственно в основной код. Однако такой подход подразумевает тесное взаимодействие с разработчиками комплекса и не всегда удобен и даже не всегда возможен. Если рассматривать комплекс NOISETTE как отчуждаемый код (который можно отдавать для использования другим коллективам), то в текущем состоянии он подразумевает, что этот коллектив будет использовать комплекс «как есть» и никак не расширять его под свои требования. А внесение в основной код своих изменений затрудняет получение новых версий.

Автором предлагается внедрить в комплекс NOISETTE новый, т.н. «модульный» подход, позволяющий решить обозначенные выше проблемы. Частично эти предложения уже приняты разработчиками основного кода. С помощью этого модульного подхода уже реализованы начальные условия, начальные условия, вязкие члены и источники. Более широкому внедрению модульного подхода препятствует то, что переход на него требует внесения существенных изменений и, соответственно, временных затрат на это, а времени, как известно, «всегда не хватает».

В докладе излагается суть модульного подхода, а также предлагаются новые идеи, в частности, т.н. «runtime class», позволяющий гибко управлять созданием и инициализацией объектов при запуске приложения.