Adaptive On-the-fly Regression Tabulation: Beyond ISAT

Edward S. Blurock*
REACTION, Lund, Sweden

Keywords

Tabulation, ISAT, Adaptive, Regression approximation.

The implementation presented in this work is an extension of the in situ adaptive tabulation (ISAT) method. In its simplest form, the method presented here mimics the ISAT method exactly. However, the method extends the ISAT method in several ways.

Reuse: In ISAT when a chemical source term point is dynamically called for, the ISAT linear approximation is calculated by systematically solving for the chemical source terms for n new points (not directly used in the calculation, i.e. n extra chemical source terms must be calculated. This could represent a severe decrement in reuse. In the method described here, the regression is not calculated until n points 'in close proximity' are called for by the calculation. Every calculation is saved and only condensed when a regression can be performed. The points are still saved for controlling the approximation.

Large Scale Accumulation: Since every calculated point is accumulated, very large scale data management must be implemented. A limitation of ISAT is that the approximation remains in memory. In the extension, an efficient swapping mechanism is established by keeping a window of a subset of tabulation is stored in memory while the total tabulation is on disk. This configuration accommodates the very large data content while still allowing for memory limitations on individual processors. In addition, it provides a natural storage of points between calculations. With each new usage, the database expands to meet new chemical source term configurations. One task of the algorithm is to ensure that data that is used more often has a higher probability to remain in memory for efficient access. In the design of data structures, particular attention has been has been made to ensure direct access, meaning low complexity by avoiding searches. If the database is smaller that the window of memory allowed by the processor, the method defaults to the traditional ISAT method.

Searching Progress Variables: A dynamically allocated and built search tree is used to access the hyper-cubes of nearest neighbors. The tree itself is not limited to binary trees. As part of the on-going development is addressing the "curse of dimensionality problem", meaning how searching for the nearest neighbor in dimensions as early as 10 progress variables cannot be addressed with traditional search methods (such as binary trees).

The algorithm itself is written in C++ and is an extension of the Cantera suite of kinetic software. In order to simulate the mixing problem, the algorithm is being tested with pairwise mixed stirred reactor (PPMSR).