Parallel methods for improving the computation of the matrix exponential using FFTs

August 31, 2004

Suppose $[0, T]$ is the interval of integration of the linear ODE problem

$$\begin{array}{rcl} y' & = & A \cdot y \\ y(0) & = & b_0, \end{array} \tag{1}$$

and consider a grid of the type $t_0 = 0 < t_1 < \cdots < t_N = T$, (the nodes will be here assumed equidistant). Suppose we want to approximate (1) by using a *solver* F, an operator which propagates the numerical solution on each of the intervals of time $[t_n, t_{n+1}]$, and $y_{n+1} = F(y_n)$. Suppose the ultimate goal is to compute at the given times the sequence of numerical approximation of the ODE solution produced by the *solver*, i.e.

$$y_0 = b_0, y_1 = F(y_0), y_2 = F(y_1), \ldots, y_N = F(y_{N-1}), \tag{2}$$

where the approximations $y_n \approx y(t_n)$, for $n = 0, \ldots, N$, have an adequate accuracy established a priori by the choice of F.

The method we propose is considering the parallel solution of (2). The idea is to consider the perturbed problem

$$y_0 = b_0 + \epsilon F(y_{N-1}), y_1 = F(y_0), y_2 = F(y_1), \ldots, y_N = F(y_{N-1}), \tag{3}$$

whose solution is as good approximation to the solution of (2) as possible. For linear constant coefficient problems $F(y_k) = F \cdot y_k$. Several choices for F are available, we can consider the exact flow where $F = \exp(hA)$ and $h = t_{n+1} - t_n$, or $F = R(hA)$ where typically R represents a polynomial or rational approximation of $\exp(hA)$, or a splitting method for the approximation of the matrix exponential.

The perturbed problem (3) is a linear system with a block-Toeplitz coefficient matrix and can now be diagonalized with the help of the Fast Fourier

Transform. Each of the diagonal-block problems can be solved separately on a different processor.

Project

The considered method should be implemented in Matlab and tested for different choices of F. The new algorithm should lead to a numerical gain in the case of exponentiation of matrices with eigenvalues of large absolute value. Analysis of the method should be performed and a parallel implementation of the algorithm might be considered.