## Extrapolation Methods: a tool for accelerating real life problems

**R. Bertelle**, M.R. Russo, M. Venturini

*Department of Pure and Applied Mathematics, University of Padua, Italy*  
*Department of Computer Science, University of Verona, Italy*

### Introduction

Electronic has a greater and greater role in nowadays life. New products, with better functionality, appear almost every day. Moreover, their lifetimes are extremely short. This requires a constant and fast development of the basic electronic devices, from which the electronic machines are made on. This low time to market necessarily needs device simulations. Fluid simulations are one of the most interesting fields since we live deep into fluids. An aircraft flies into air which may be seen as an inviscid compressible fluid. A wind power plant uses wind to produce electricity. Water, an incompressible fluid, flows inside aqueducts from the spring to our houses. Also our blood may be modeled as a complicated uid. All these aspects may be optimized and understood better via computer simulations.

The convergence of the simulation algorithms is sometimes very slow. In such cases is essential to improve the rate of convergence and this can be done by vector extrapolation methods.

### Vector Extrapolation Methods

Vector extrapolation methods accelerate slowly or non convergent vector sequences. We consider here the Polynomial Extrapolation methods, which are a sequence of N-dimensional column vectors with limit s. A polynomial algorithm finds s as a weighted average of k + 1 terms related with the given sequence, where the independent weights are found by solving a linear system of size (N × k). In detail this means:

\[ U = [u_0, u_1, \ldots, u_k] \]

\[ \gamma_k = \gamma_k^{(0)} + \gamma_k^{(1)} s \]

MPE-Minimal Polynomial Extrapolation:

\[ s = \sum_{j=1}^{k} \gamma_j v_j \]

where the coefficients \( \gamma_j \), \( j = 0, 1, \ldots, k \) are obtained solving the linear system \( C = \gamma_0 u_0 \).

RRE-Reduced Rank Extrapolation:

\[ s = x_0 + \sum_{j=1}^{k} \beta_j \]

where the coefficients \( \beta_j \), \( j = 0, 1, \ldots, k \) are obtained solving the linear system \( \gamma_0 s = -u_0 \).

**Remarks**
- The explicit knowledge of the sequence generator is not required.
- \( s \) is found solving a relatively small linear system (i.e., \( k \ll N \)).

### Extrapolation of the Gummel map

The model of a semiconductor device is a system of three strongly coupled differential equations:

- The Poisson equation:
  \[ \hat{\phi} = \nabla \cdot \rho = \frac{\partial \psi}{\partial x} \]
- The continuity equations:
  \[ \frac{\partial u}{\partial x} = 0 \]
- The quasi-Fermi potentials:
  \[ \frac{\partial \psi}{\partial x} = \frac{\partial \psi}{\partial x} \]

\[ \psi = \frac{\psi^0}{\psi^1} \]

\[ \psi = \frac{\psi^0}{\psi^1} \]

We propose an accelerated version of the algorithm.

### Extrapolation of CBS scheme

There are no general guidelines to accelerate the convergence of evolutive problems that converge to a stationary solution. So, we adopt the following strategies:
- We select the variables to check where the extrapolation procedure should be done. Regardless this choice, all the solution variables are extrapolated.
- We skip the first iterations of the iterative process since they are, reasonable, the most far away from the stationary solution.
- We collect consecutive iterations characterized by the fact that the maximum error for each variable between two consecutive iterations occurs at the same mesh nodes. The errors between consecutive iterations are proportional and hence it is reasonably that it appears at the same nodes of the mesh. This way to check the extrapolation point is very less, memory, and time consuming.
- We perform extrapolation only on the last part of the collected iterations.
- Finally, since the extrapolated solution, typically, is not a feasible solution for the evolution problem (i.e., it does not satisfy the PDE equations), we give some iterations to relax towards a feasible solution before looking back again to original sequence for finding another starting point before collecting further iterations.

### Buoyancy-Driven Convection Problem

In this problem the Navier-Stokes equations are coupled with the temperature equation. The local temperature difference induces a local density difference within the fluid and this produce the fluid motion.

### NACA0012 Airfoil Problem

This problem refers to an inviscid compressible flow over a NACA0012 airfoil test problem. At Mach number 0.25.

### Fluid simulations

The mathematical model for both compressible and incompressible fluids contains continuity and momentum equations and is completed in the compressible fluids by a temperature equation whereas for the incompressible ones by an energy equation.

The numerical solution may be carried out using an iterative procedure, the Characteristic-Based Split (CBS) scheme. It is a fractional time-stepping algorithm based on an original finite difference velocity-projection scheme where the convective terms are treated using the idea of the characteristic-Galerkin method. The number of iteration may be very high and thus we propose an accelerated version of the algorithm.

### References


