Understanding the rational approximation of the exponential integrator (REXI)

Martin Schreiber <M.Schreiber@exeter.ac.uk>

July 27, 2015

This document serves as the basis for implementing the rational approximation of the exponential integrator (REXI). Here, we purely focus on the linear part of the shallow-water equations (SWE) and show the different steps to approximate solving this linear part with an exponential integrator. This paper is mainly summarises previous work on REXI.

1 Problem formulation

We use the advective formulation of the SWE with a full linearization with perturbation (see [1]) with $U := (h, u, v)^T$, yielding

$$L(U) := \begin{pmatrix} & H\delta_x & H\delta_y \\ g\delta_x & & -f \\ g\delta_y & f & \end{pmatrix} U$$

Here, we neglect all non-linear terms, assume only small perturbations (hence negligible ones) around the average surface height and negligible non-linear terms.

Then, the time evolution of the PDE with the subscript t denoting the derivative in time is given by

$$U_t := L(U).$$

We continue writing the linear operator in matrix-style L and applying L on U as $L.U$.

$$U_t := L.U$$

It is further worth noting, that this system describes an oscillatory system, hence the operator L has imaginary eigenvalues.

2 Exponential integrator

Linear systems of equations are well known to be solvable with exponential integrators for arbitrary time step sizes via

$$U(t) := e^{Lt} U(0).$$

However, this is typically quite expensive to compute and analytic solutions only exist for some simplified system of equations, see e.g. [1] for f-plane shallow-water equations. These exponential integrators can be approximated with rational functions and this paper is on giving insight into this approximation.

3 Underlying idea of rational approximation

Terry et. al. developed a rational approximation of the exponential integrator, see [2]. First, we like to get more insight into it with a one-dimensional formulation before applying REXI to a rational approximation of a linear operator. Our main target is to find an approximation of an operator with an *exponential shape*, such as e^{ix}, which (in one-dimension) is given by a function $f(x)$.

3.1 Step A) Approximation of solution space

First, we assume that we can use Gaussian curves as basis functions for our approximation and they are somehow naturally related to this problem (Gaussians are given by exponential functions). So first, we find an approximation of one of our underlying Gaussian basis function

$$\psi_s(x) := (4\pi)^{-\frac{1}{2}} e^{-x^2/(4s^2)}$$

In this formulation, s can be interpreted as the horizontal "stretching" of the basis function. Note the similarities to the Gaussian distribution, but by dropping certain parts of the vertical scaling as it is required for probability distributions. We can now approximate our function $f(x)$ with the Gaussian distributions:

$$f(x) \approx \sum_{m=-M}^{M} b_m \psi_s(x + ms)$$

with M controling the interval of approximation (~size of "domain of interest") and s the accuracy of integration (~resolution in "domain of interest").

3.2 Step B) Approximation of basis function

The second step is the approximation of the basis function $\psi_s(x)$ itself via

$$\psi_s(x) \approx 2Re\left(\sum_{l=-L}^{L} \frac{a_j}{ix - (\mu + i\,l)}\right).$$

See the paper for these mathematical magic tricks.... The weights are derived via optimization in Fourier space.

3.3 Step C) Approximation of the approximation

We then combine the approximation (B) of the approximation (A), yielding

$$f(x) \approx 2Re\left(\sum_{n=-M-L}^{M+L} \frac{c_n}{ix - s(\mu + i\,n)}\right)$$

which gives us the REXI for a one-dimensional function with precomputed values of c_n and μ.

4 REXI, our little dog

In the following, we use $L := \tau L'$ and assume an a-priori fixed time step size, making a REXI approximation more efficient. Then, the REXI approximation is given by

$$exp(\tau L') \approx \sum_{k=-K}^{K} \beta_k (L - \alpha_k)^{-1} \tag{1}$$

The coefficients α_k (corresponding to $s(\mu + i\,n)$ in step C for the one-dimensional formulation) can be precomputed based on a one-dimensional approximation, see the paper, and can be interpreted as shifts of the rational approximations. The coefficients β_k (corresponding to c_n in step C) are describing the scaling of the basis function and depend on the solution itself.

Note an important property (see Sec. 3.3 in [2]). There's an anti-symmetry in the α_i coefficients, which avoids computing half of the inverses:

$$\overline{(L - \alpha)^{-1} u_0} = (L - \overline{\alpha})^{-1} u_0$$

5 Computing inverse of $(L - \alpha)^{-1}$

We still have to compute the inverse of $(L - \alpha)$ which can be very expensive if directly computing it. Here, we consider a specialization on the shallow-water equations given above with

$$L(U) := \begin{pmatrix} & H\delta_x & H\delta_y \\ g\delta_x & & -f \\ g\delta_y & f & \end{pmatrix} U,$$

$$U_t := L(U)$$

and we set $g := 1$ and the average height $H := 1$. For the REXI method, we now have to solve the systems of equations given by

$$(L - \alpha).U = U0$$

with U_0 the initial conditions. According to [4], instead of solving this relatively large system of equations we can split the problem into an elliptic one for the height which then allows to use an explicit formulation for the velocities. We use the abbreviation$\vec{v} := (u, v)$ in the following paragraph. Using the formulation in [2], the height can be computed with the elliptic equation given by

$$(\nabla^2 - (\alpha^2 + f^2))h(\tau) = \frac{\alpha^2 + f^2}{\alpha}(h(0) + H\nabla \cdot (A.v(0)) \tag{2}$$

with

$$A := \frac{1}{\alpha^2 + f^2} \begin{pmatrix} \alpha & -f \\ f & \alpha \end{pmatrix}.$$

We can rearrange this equation by using the abbreviations $\kappa := \alpha^2 + f^2$ and $\gamma := \alpha^{-1}$ in the following way:

$$(\nabla^2 - \kappa)\, h(\tau) = \frac{\kappa}{\alpha}\,(h(0) + H\nabla \cdot (A.v(0))$$

$$(\nabla^2 - \kappa)\, h(\tau) = \frac{\kappa}{\alpha}h(0) + \frac{1}{\alpha}H\nabla \cdot \begin{pmatrix} \alpha & -f \\ f & \alpha \end{pmatrix}.v(0)$$

$$(\nabla^2 - \kappa)\, h(\tau) = \frac{\kappa}{\alpha}h(0) + \frac{1}{\alpha}H \begin{pmatrix} \alpha & -f \\ f & \alpha \end{pmatrix} \nabla \cdot v(0)$$

$$(\nabla^2 - \kappa)h(\tau) = \frac{\kappa}{\alpha}h(0) - \frac{Hf}{\alpha}\nabla \times v(0) + H\nabla \cdot v(0) \tag{3}$$

Here, the α denote the only terms with imaginary numbers. On the right hand side, we see an update-like scheme $h(0)$ in the first scheme, then a vorticity-like formulation $\times$, and an advection part ∇ of the height. Once computed the height, the velocities can be directly computed via

$$\vec{v}(\tau) = -A.\vec{v}(0) + A.\nabla h$$

$$\vec{v}(\tau) = -A.(\vec{v}(0) + \nabla h) \tag{4}$$

giving us our final solution

$$U := (h, v, u)^T.$$

One final scaling has to be done: the exponential is computing $e^{\tau L}$, hence the τ has to be included in the operator L. There are basically two different ways: The first one is rescaling all parameters by τ:

$$g' := \tau g$$

$$f' := \tau f$$

$$h'_0 := \tau h_0$$

The second way is to factor the τ parameter out:

$$(\tau L - \alpha).U(\tau) = U0$$

$$(L - \frac{\alpha}{\tau}).U(\tau)\tau = U0$$

So instead of solving for $U(\tau)$, we are solving for $U(\tau)\tau$ and have to divide the computed solution by τ in the end.

6 Bringing everything together

Using the spectral methods (e.g. in SWEET), we can directly solve the height Eq. (2) and then solver for the velocity in Eq. (4). Then, the problem is reduced to computing the REXI as given in Eq. (1). We like to note again, that the α_i are independent of the system L to solve, and only the β_i depend on the system L.

7 Acknowledgements

Thanks to Pedro & Terry!

References

[1] Formulations of the shallow-water equations, M. Schreiber

[2] High-order time-parallel approximation of evolution operators, T. Haut et. al.

[3] An asymptotic parallel-in-time method for highly oscillatory PDEs, T. Haut et. al.

[4] An invariant theory of the linearized shallow water equations with rotation and its application to a sphere and a plane, N. Paldor et. al.