Exponential Time Differencing Explicit Runge-Kutta Methods

January 7, 2020

We study ODEs of type

$$\frac{\mathrm{d}}{\mathrm{d}t} y(t) = \mathbf{L}\mathbf{y}(t) + \mathbf{N}(y(t), t), \tag{1}$$

where y may be a vector, L is a linear operator, and N is a nonlinear operator. If L has a wide range of eigenvalues, the above ODE is *stiff*. Exponential time differencing methods were designed to treat stiff ODEs.

1 Exponential Time Differencing (ETD) Methods

We make the same change of variable as we do in the Integrating Factor (IF) methods. We can rewrite Equation 1 as the following.

$$\frac{\mathrm{d}}{\mathrm{d}t} \left(\exp(-t\mathbf{L})y\right) = \exp(-t\mathbf{L})\mathbf{N}(\mathbf{y}, t) \tag{2}$$

Given a state $\mathbf{y}_n := y(t_n)$, we can employ this change of variables at that time to move forward in time by h.

$$\mathbf{y}_{n+1} = \exp(h\mathbf{L}) \left(\mathbf{y}_n + \int_0^h \exp(-\tau\mathbf{L})\mathbf{N}(y(t_n + \tau), t_n + \tau)\mathrm{d}\tau \right) \tag{3}$$

This equation is exact. All ETD methods derive from approximating the integral.

1.1 Multi-step Explicit ETD methods

Some basic ETD methods work by approximating the nonlinear function in the integrand in Equation 3 using previous steps. Here, the task is to approximate $\mathbf{N}(\mathbf{y}, t) = \mathbf{N}(y(t), t)$ only for values of t in $[t_n, t_n + h]$. A constant approximation of $\mathbf{N}(y)$ by using $\mathbf{N}(\mathbf{y}_n)$) yields ETD1, and a linear approximation using $\mathbf{N}(\mathbf{y}_n)$ and $\mathbf{N}(\mathbf{y}_{n-1})$ yields ETD2.

$$\begin{aligned}
&\text{ETD1} : \mathbf{N}(y(t), t) \approx \mathbf{N}(\mathbf{y}_n) \\
&\text{ETD2} : \mathbf{N}(y(t), t) \approx \mathbf{N}(\mathbf{y}_n) + \tau \frac{\mathbf{N}(\mathbf{y}_n) - \mathbf{N}(\mathbf{y}_{n-1})}{h}
\end{aligned}$$

We solve the integral exactly with these constant and linear approximations to get the full schemes, which are shown below.

$$\text{ETD1}: \mathbf{y}_{n+1} = \exp(h\mathbf{L})\mathbf{y}_n + \mathbf{N}(\mathbf{y}_n)\frac{\exp(h\mathbf{L}) - 1}{L}$$
$$\text{ETD2}: \mathbf{y}_{n+1} = \exp(h\mathbf{L})\mathbf{y}_n + \mathbf{N}(\mathbf{y}_n)\frac{(1 + h\mathbf{L})\exp(h\mathbf{L}) - 2h\mathbf{L} - 1}{h\mathbf{L}^2} + \mathbf{N}(\mathbf{y}_{n-1})\frac{-\exp h\mathbf{L} + 1 + h\mathbf{L}}{h\mathbf{L}^2}$$

If L is a scalar linear operator, the above equations make sense, an if it is not, then 1's should be replaced by identity matrices.

1.2 Multi-stage methods: Runge-Kutta

Runge-Kutta methods with s-stages work by approximating the state, y, and its time-derivative, $\frac{\mathrm{d}}{\mathrm{d}t}y$, in various stages,$\{t_n + c_i h\}_{i=1}^s$, where $c_i \in [0, h]$ within a single time-step, h. Instead, ETD-RK methods only approximate the nonlinear function in the integrand of Equation 3 (instead of the whole RHS).

1.2.1 ETD-RK2

Consider Heun's second order scheme shown by Butcher tableau below.

$$\begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

This scheme has two stages which both approximate $\mathbf{y}_{n+1}$

$$\mathbf{y}_n = \mathbf{k}_1 := \mathbf{y}_n \tag{4}$$
$$\mathbf{y}_{n+1} \approx \mathbf{k}_2 := \exp(h\mathbf{L})\mathbf{y}_n + \mathbf{N}(\mathbf{y}_n)\frac{\exp(h\mathbf{L}) - 1}{L} \quad \text{(same as ETD1)} \tag{5}$$
$$\mathbf{N}(\mathbf{y}, t) \approx N_{RK2}(\mathbf{k}_1, \mathbf{k}_2) := \mathbf{N}(\mathbf{y}_n, t_n) + \tau\frac{\mathbf{N}(\mathbf{k}_2) - \mathbf{N}(\mathbf{k}_1)}{h} \tag{6}$$

Integrating exactly with this linear approximation yields ETD-RK2.

$$\mathbf{y}_{n+1} = \mathbf{k}_2 + (\mathbf{N}(\mathbf{k}_2) - \mathbf{N}(\mathbf{y}_n))\frac{\exp(h\mathbf{L}) - h\mathbf{L} - 1}{h\mathbf{L}^2} \tag{7}$$

So the name of the game is to find approximations of the nonlinear term in the integrand. If we stick to polynomial approximations with interpolating nodes within a time-step, we can make the generalization that the resulting ETD-RK schemes all involve integrating $\exp(h\mathbf{L})\tau^i$.

Suppose that the RK scheme yields some polynomial approximant,

$$\mathbf{N}(y) = \mathbf{N}(y(t_n + \tau)) \approx \mathbf{N}_{RK}(\tau) = p_0 + p_1\tau + \cdots p_s\tau^s.$$

Then the exact integral we must solve can now be separated by each term in the polynomial.

$$\int_0^h \exp(-\tau\mathbf{L})\mathbf{N}(y(t_n + \tau), t_n + \tau)\mathrm{d}\tau \approx \sum_{i=0}^s p_i \int_0^h \exp(-\tau\mathbf{L})\tau^i\mathrm{d}\tau \tag{8}$$

Substituting this approximation into the general ETD scheme yields,

$$\mathbf{y}_{n+1} = \exp(h\mathbf{L})\left(\mathbf{y}_n + \sum_{i=0}^{s} p_i \int_0^h \exp(-\tau\mathbf{L})\tau^i \mathrm{d}\tau\right) \tag{9}$$

ϕ functions: Here we introduce a family of functions that deal with these sorts of integrals. We have $\phi_0(h\mathbf{L}) := \exp h\mathbf{L}$, and for $l \geq 1$,

$$\phi_l(h\mathbf{L}) = \frac{\exp(h\mathbf{L})}{h^l(l-1)!}\int_0^h \exp(-\tau\mathbf{L})\tau^{l-1}\mathrm{d}\tau. \tag{10}$$

Let's define new coefficients q_l's, so that we can write the scheme in terms of the ϕ_l functions. The indexing is off, so think $l = i+1$. We have to require $q_l = (l-1)!h^l p_{l-1}$ for $l \geq 1$.

$$\mathbf{y}_{n+1} = \exp(h\mathbf{L})\mathbf{y}_n + \sum_{l=1}^{s+1} q_l \phi_l(h\mathbf{L}) \tag{11}$$

ETDRK2 represented by ϕ functions: Equation 6 is simply a linear approximation with $p_0 = \mathbf{N}(\mathbf{y}_n)$ and $p_1 = \frac{\mathbf{N}(\mathbf{k}_2)-\mathbf{N}(\mathbf{y}_n)}{h}$. This gives us $s = 1$, $q_1 = h\mathbf{N}(\mathbf{y}_n)$, and $q_2 = h^2\frac{\mathbf{N}(\mathbf{k}_2)-\mathbf{N}(\mathbf{y}_n)}{h}$. Using this, we can construct a Butcher tableau like before.

$$\begin{array}{c|cc} 0 & 0 & \\ 1 & \phi_1(h\mathbf{L}) & 0 \\ \hline & \phi_1(h\mathbf{L}) - \phi_2(h\mathbf{L}) & \phi_2(h\mathbf{L}) \end{array} = \begin{array}{c|cc} 0 & 0 & \\ c_2 & a_{21} & 0 \\ \hline & b_1 & b_2 \end{array}$$

2 Tableau

How to interpret these ETD-RK tableaus as is implied in Equation 5.4 of [Minchev and Wright, 2005]:

$$\mathbf{k}_i = \exp(c_i h\mathbf{L})\mathbf{y}_n + h\sum_{j=1}^{i-1} a_{ij}\mathbf{N}(\mathbf{k}_j) \tag{12}$$

$$\mathbf{y}_{n+1} = \exp(h\mathbf{L})\mathbf{y}_n + h\sum_{j=1}^{s} b_j\mathbf{N}(\mathbf{k}_j) \tag{13}$$

Furthermore, the a_{ij}'s and b_j's are defined via ϕ functions in the following way, (Equation 5.5 of [Minchev and Wright, 2005]), which was originally published in [Friedli, 1978].

$$a_{ij} = \sum_{k=1}^{i-1} \alpha_{ijk}\phi_k(c_i h\mathbf{L})$$

$$b_i = \sum_{k=1}^{s} \beta_{ik}\phi_k(h\mathbf{L})$$

ETDRK3

$$
\begin{array}{c|ccc}
0 & & & \\
\frac{1}{2} & \frac{1}{2}\phi_1(\frac{h}{2}\mathbf{L}) & & \\
1 & -\phi_1(h\mathbf{L}) & 2\phi_1(h\mathbf{L}) & \\
\hline
 & \phi_1(h\mathbf{L}) - 3\phi_2(h\mathbf{L}) + 4\phi_3(h\mathbf{L}) & 4\phi_2(h\mathbf{L}) - 8\phi_3(h\mathbf{L}) & -\phi_2(h\mathbf{L}) + 4\phi_3(h\mathbf{L})
\end{array}
$$

ETDRK4

$$
\begin{array}{c||c|c|c|c|}
0 & & & & \\
\frac{1}{2} & \frac{1}{2}\phi_1(\frac{h}{2}\mathbf{L}) & & & \\
\frac{1}{2} & 0 & \frac{1}{2}\phi_1(\frac{h}{2}\mathbf{L}) & & \\
1 & \frac{1}{2}\phi_1(\frac{h}{2}\mathbf{L})(\phi_0(\frac{h}{2}\mathbf{L}) - 1) = \frac{h}{4}L[\phi_1(\frac{h}{2}\mathbf{L})]^2 & 0 & \phi_1(\frac{h}{2}\mathbf{L}) & \\
\hline
 & \phi_1(h\mathbf{L}) - 3\phi_2(h\mathbf{L}) + 4\phi_3(h\mathbf{L}) & 2\phi_2(h\mathbf{L}) - 4\phi_3(h\mathbf{L}) & 2\phi_2(h\mathbf{L}) - 4\phi_3(h\mathbf{L}) & -\phi_2(h\mathbf{L}) + 4\phi_3(h\mathbf{L})
\end{array}
$$

ETDRK4-B :What's printed on (51) of [Krogstad, 2005].

$$
\begin{array}{c||c|c|c|c|}
0 & & & & \\
\frac{1}{2} & \frac{1}{2}\phi_1 & & & \\
\frac{1}{2} & \frac{1}{2}\phi_1 - \phi_2 & \phi_2 & & \\
1 & \phi_1 - 2\phi_2 & 0 & 2\phi_2 & \\
\hline
 & \phi_1 - 3\phi_2 + 4\phi_3 & 2\phi_2 - 4\phi_3 & 2\phi_2 - 4\phi_3 & -\phi_2 + 4\phi_3
\end{array}
$$

ETDRK4-B :What Ian's matlab code suggests.

$$
\begin{array}{c||c|c|c|c|}
0 & & & & \\
\frac{1}{2} & \frac{1}{2}\phi_1(\frac{h}{2}\mathbf{L}) & & & \\
\frac{1}{2} & \frac{1}{2}\phi_1(\frac{h}{2}\mathbf{L}) - \phi_2(\frac{h}{2}\mathbf{L}) & \phi_2(\frac{h}{2}\mathbf{L}) & & \\
1 & \phi_1(h\mathbf{L}) - 2\phi_2(h\mathbf{L}) & 0 & 2\phi_2(h\mathbf{L}) & \\
\hline
 & \phi_1(h\mathbf{L}) - 3\phi_2(h\mathbf{L}) + 4\phi_3(h\mathbf{L}) & 2\phi_2(h\mathbf{L}) - 4\phi_3(h\mathbf{L}) & 2\phi_2(h\mathbf{L}) - 4\phi_3(h\mathbf{L}) & -\phi_2(h\mathbf{L}) + 4\phi_3(h\mathbf{L})
\end{array}
$$

3 Resonant Triad Analysis: ETD-Explicit RTD methods

We can write the resonant triad equations in the form of Equation 1,

$$
\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{z} = \begin{bmatrix} \mathrm{i}\omega_1 & 0 & 0 \\ 0 & \mathrm{i}\omega_2 & 0 \\ 0 & 0 & \mathrm{i}\omega_3 \end{bmatrix} \mathbf{z} + \epsilon \begin{bmatrix} C_1 z_2^* z_3^* \\ C_2 z_1^* z_3^* \\ C_3 z_1^* z_2^* \end{bmatrix} = \mathbf{L}\mathbf{z} + \mathbf{N}(\mathbf{z}),
$$

where $\mathbf{z} = [z_1, z_2, z_3]^\top$.

3.1 Order ϵ perturbation to argument of nonlinear function.

First, we show what happens when we add an order ϵ perturbation to the argument of $\mathbf{N}(\cdot)$. Let $\mathbf{z} = \mathbf{z}^0 + \epsilon\mathbf{z}^{(1)}$, where $\mathbf{z}^{(0)}, \mathbf{z}^{(1)} = \mathcal{O}(1)$. Then,

$$\mathbf{N}(\mathbf{z}^0 + \epsilon\mathbf{z}^{(1)}) = \epsilon \begin{bmatrix} C_1(z_2^{(0)} + \epsilon z_2^{(1)})^*(z_3^{(0)} + \epsilon z_3^{(1)})^* \\ C_2(z_1^{(0)} + \epsilon z_1^{(1)})^*(z_3^{(0)} + \epsilon z_3^{(1)})^* \\ C_3(z_1^{(0)} + \epsilon z_1^{(1)})^*(z_2^{(0)} + \epsilon z_2^{(1)})^* \end{bmatrix}$$

$$= \epsilon \begin{bmatrix} C_1 z_2^{(0)*} z_3^{(0)*} \\ C_2 z_1^{(0)*} z_3^{(0)*} \\ C_3 z_1^{(0)*} z_2^{(0)*} \end{bmatrix} + \epsilon^2 \begin{bmatrix} C_1 \left(z_2^{(0)*} z_3^{(1)*} + z_2^{(1)*} z_3^{(0)*} \right) \\ C_2 \left(z_1^{(0)*} z_3^{(1)*} + z_1^{(1)*} z_3^{(0)*} \right) \\ C_3 \left(z_1^{(0)*} z_2^{(1)*} + z_1^{(1)*} z_2^{(0)*} \right) \end{bmatrix} + \epsilon^3 \begin{bmatrix} C_1 z_2^{(1)*} z_3^{(1)*} \\ C_2 z_1^{(1)*} z_3^{(1)*} \\ C_3 z_1^{(1)*} z_2^{(1)*} \end{bmatrix} = \mathbf{N}(\mathbf{z}^{(0)}) + \mathcal{O}(\epsilon^2)$$

3.2 Apply to ETD explicit RK methods on the resonant triad

We apply the above result to RK methods described by Equations 12 through 13. **Claim:** For the resonant triad equations, $\mathbf{N}(\mathbf{k}_i) = \exp(-c_i h\mathbf{L})\mathbf{N}(\mathbf{y}_{n-1}) + \mathcal{O}(\epsilon^2)$.

Proof. We prove this via induction.
i = 1:
Recall that $\mathbf{k_1} = \exp(c_1 h\mathbf{L})\mathbf{y}_{n-1}$. Then,

$$\mathbf{N}(\mathbf{k}_1) = \mathbf{N}(\exp(c_1 h\mathbf{L})\mathbf{y}_{n-1}) = \epsilon \begin{bmatrix} C_1 \exp(c_1 h\omega_1) y_{n-1,2}^* y_{n-1,3}^* \\ C_2 \exp(c_1 h\omega_2) y_{n-1,1}^* y_{n-1,3}^* \\ C_3 \exp(c_1 h\omega_3) y_{n-1,1}^* y_{n-1,2}^* \end{bmatrix} = \exp(c_1 h\mathbf{L})\mathbf{N}(\mathbf{y}_{n-1}).$$

Induction step: Suppose that $\mathbf{N}(\mathbf{k}_j) = \exp(c_j h\mathbf{L})\mathbf{N}(\mathbf{y}_{n-1}) + \mathcal{O}(\epsilon^2)$ for all $j = 1, \cdots, i-1$. Then,

$$\begin{aligned} \mathbf{k}_i &= \exp(c_i h\mathbf{L})\mathbf{y}_{n-1} + h\sum_{j=1}^{i-1} a_{ij}(h\mathbf{L})\mathbf{N}(\mathbf{k}_j) \\ &= \exp(c_i h\mathbf{L})\mathbf{y}_{n-1} + h\sum_{j=1}^{i-1} a_{ij}(h\mathbf{L})\left[\exp(c_j h\mathbf{L})\mathbf{N}(\mathbf{y}_{n-1}) + \mathcal{O}(\epsilon^2)\right] \end{aligned}$$

We know that $\mathbf{N}(\mathbf{y}_{n-1}) = \mathcal{O}(\epsilon)$, so we also know that the leading term in the summand, $a_{ij}(h\mathbf{L})\exp(c_j h\mathbf{L})\mathbf{N}(\mathbf{y}_{n-1})$, is order ϵ as well. Using this, we can add further simplify the asymptotic ordering.

$$\begin{aligned} \mathbf{N}(\mathbf{k}_i) &= \mathbf{N}\Bigg(\exp(c_i h\mathbf{L})\mathbf{y}_{n-1} + h\sum_{j=1}^{i-1} a_{ij}(h\mathbf{L})\left[\exp(c_j h\mathbf{L})\mathbf{N}(\mathbf{y}_{n-1}) + \mathcal{O}(\epsilon^2)\right] \Bigg) \\ &= \mathbf{N}\Bigg(\exp(c_i h\mathbf{L})\mathbf{y}_{n-1} + h\sum_{j=1}^{i-1} a_{ij}(h\mathbf{L})\exp(c_j h\mathbf{L})\mathbf{N}(\mathbf{y}_{n-1}) + \mathcal{O}(h(i-1)\epsilon^2) \Bigg) \\ &= \mathbf{N}(\exp(c_i h\mathbf{L})\mathbf{y}_{n-1}) + \mathcal{O}(\epsilon^2) \\ &= \exp(c_i h\mathbf{L})\mathbf{N}(\mathbf{y}_{n-1}). \end{aligned}$$

Note that this last step is specific to the resonant triad. □

Plugging in this result into Equation 13:

$$\begin{aligned}
\mathbf{y}_n &= \exp(h\mathbf{L})\mathbf{y}_{n-1} + h\sum_{j=1}^{s} b_j \mathbf{N}(\mathbf{k}_j) \\
&= \exp(h\mathbf{L})\mathbf{y}_{n-1} + h\sum_{j=1}^{s} b_j(h\mathbf{L})[\exp(c_j h\mathbf{L})\mathbf{N}(\mathbf{y}_{n-1}) + \mathcal{O}(\epsilon^2)] \\
&= \exp(h\mathbf{L})\mathbf{y}_{n-1} + h\sum_{j=1}^{s} b_j(h\mathbf{L})\exp(c_j h\mathbf{L})\mathbf{N}(\mathbf{y}_{n-1}) + \mathcal{O}(\epsilon^2)
\end{aligned}$$

References

[Friedli, 1978] Friedli, A. (1978). Verallgemeinerte Runge–Kutta Verfahren zur Lösung steifer Differentialgleichungssysteme. *Numerical treatment of differential equations*, 631:35–50.

[Krogstad, 2005] Krogstad, S. (2005). Generalized integrating factor methods for stiff PDEs. *Journal of Computational Physics*, 203(1):72–88.

[Minchev and Wright, 2005] Minchev, B. and Wright, W. (2005). A review of exponential integrators for first order semi-linear problems. *Preprint Numerics*, 2:1–45.