

# Applications for Mixed-precision HQR

August 20, 2019

## 1 Discovery of Equations

There is growing interest in data-driven “discovery of equations” (c.f. [1]). The simplest example finds an autonomous system of ODEs given noisy data of the state variables. Consider some ODE,

$$\dot{\mathbf{x}} = \vec{f}(\mathbf{x}), \quad (1)$$

where  $\mathbf{x}, \frac{d}{dt}\mathbf{x} \in \mathbb{R}^m$ , and  $\vec{f}: \mathbb{R}^m \rightarrow \mathbb{R}^m$  may be linear or nonlinear. Suppose further that

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix},$$

and we have at our disposal time-series data of these variables,

$$\begin{aligned} \mathbf{X} &= [\mathbf{X}_1, \dots, \mathbf{X}_n]^\top \\ \mathbf{Y} &= [\mathbf{Y}_1, \dots, \mathbf{Y}_n]^\top \\ \mathbf{Z} &= [\mathbf{Z}_1, \dots, \mathbf{Z}_n]^\top. \end{aligned}$$

Our goal is to identify the function  $\vec{f}$  as some function of  $x$ ,  $y$ , and  $z$ . Let’s also assume that the data was collected at regular time intervals. The approach suggested in [1] is as follows.

Step 1. Generate the left hand side, the time derivative, from the data. This can be done with any finite difference scheme or more complex schemes, and we show a 2nd order central difference scheme,

$$\begin{aligned} \dot{\mathbf{X}} &= (\mathbf{X}[3:n] - \mathbf{X}[1:n-2])/(2\Delta t) \\ \dot{\mathbf{Y}} &= (\mathbf{Y}[3:n] - \mathbf{Y}[1:n-2])/(2\Delta t) \\ \dot{\mathbf{Z}} &= (\mathbf{Z}[3:n] - \mathbf{Z}[1:n-2])/(2\Delta t). \end{aligned}$$

Step 2. Generate a *library* of terms from the data.

$$\Theta(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = \begin{bmatrix} \mathbf{X}_2 & \mathbf{Y}_2 & \mathbf{Z}_2 & \mathbf{X}_2^2 & \mathbf{X}_2\mathbf{Y}_2 & \dots \\ \mathbf{X}_3 & \mathbf{Y}_3 & \mathbf{Z}_3 & \mathbf{X}_3^2 & \mathbf{X}_3\mathbf{Y}_3 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ \mathbf{X}_{n-1} & \mathbf{Y}_{n-1} & \mathbf{Z}_{n-1} & \mathbf{X}_{n-1}^2 & \mathbf{X}_{n-1}\mathbf{Y}_{n-1} & \dots \end{bmatrix}$$

Step 3. Solve for  $\mathbf{w}_x, \mathbf{w}_y, \mathbf{w}_z$  where,

$$\dot{\mathbf{X}} = \Theta(\mathbf{X}, \mathbf{Y}, \mathbf{Z})\mathbf{w}_x \quad (2)$$

$$\dot{\mathbf{Y}} = \Theta(\mathbf{X}, \mathbf{Y}, \mathbf{Z})\mathbf{w}_y \quad (3)$$

$$\dot{\mathbf{Z}} = \Theta(\mathbf{X}, \mathbf{Y}, \mathbf{Z})\mathbf{w}_z. \quad (4)$$

$$(5)$$

In Step 3, we may use any type of linear regression techniques, although we may want to use to our advantage the following facts.

- The library should contain as many possible terms as possible because we need the true solution to be some linear combination of the terms in the library.
- However, the solution probably contains much fewer terms than what is included in the library.

While methods such as LASSO regularization encourages sparse solutions, [1] suggests another simple approach that can be added to LASSO and other methods. Suppose that we start with a large library,  $\Theta(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ . With some tolerance value,  $\epsilon$ , we repeat until some stopping criterion:

Step 1. Solve for  $\mathbf{w}$ .

Step 2. Update  $\mathbf{w}$  by dropping all indices of  $\mathbf{w}$  that are smaller than  $\epsilon$ .

Step 3. Update  $\Theta(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$  by dropping the corresponding columns.

This requires multiple linear regression solves where the accuracy of each solve only matters in whether the weights fall below or above the tolerance value,  $\epsilon$ .

## 1.1 Lower precision HQR as Linear Solve

The procedure described in the previous section is iterative, and it appears that high precision may not be necessary in each iteration. This motivates the use of mixed-precision HQR as the linear solve in the procedure, and Section 1.1.1 shows details and results from our experiments. However, the authors in [1] now suggest a different approach for sparse relaxed regularized regression, called SR3 (c.f. [2]). This method seems to rely on conjugate gradient based implementation, which I think might be worth looking into in a mixed-precision setting since they also heavily rely on inner products.

### 1.1.1 Experiment Details

To test whether mixed-precision HQR is a good candidate for this application, we picked the Lorenz system (Equations 6 - 8) as our test ODE system, and generated data via a generic ODE solver.

$$\dot{x} = \sigma(y - x) \quad (6)$$

$$\dot{y} = x(\rho - z) - y \quad (7)$$

$$\dot{z} = xy - \beta z \quad (8)$$

We used a fourth-order central finite difference scheme (Equation 9) and added an iid  $\alpha$  standard normal sample, where  $\alpha$  is the noise amplitude to generate  $\dot{\mathbf{X}}$ ,  $\dot{\mathbf{Y}}$ , and  $\dot{\mathbf{Z}}$ .

$$\dot{u}(t)|_{t=i} = \frac{-u_{i+2} + 8u_{i+1} - 8u_{i-1} + u_{i-2}}{12\Delta t} + \mathcal{O}(\Delta t^4) \quad (9)$$

Some preliminary results show that same accuracy can be achieved from mixed-precision and single precision QR solves as the double-precision QR can produce, especially in simulated data with high noise.

### 1.1.2 Regularization

I only tried L2-norm regularization (Tikhonov) since it is easy to implement on a matrix linear solve. However, it is the L1-norm regularization (LASSO) that induces sparsity.

## 2 Fiedler Value

## References

- [1] S. L. Brunton, J. L. Proctor, and J. N. Kutz, “Discovering governing equations from data by sparse identification of nonlinear dynamical systems,” *Proceedings of the National Academy of Sciences*, vol. 113, no. 15, pp. 3932–3937, 2016.
- [2] P. Zheng, T. Askham, S. L. Brunton, J. N. Kutz, and A. Y. Aravkin, “A unified framework for sparse relaxed regularized regression: Sr3,” *IEEE Access*, vol. 7, pp. 1404–1423, 2018.