# Matrix-Vector Representation of Discretization in 1D Linear Advection Equation

This is a demo to show how the for loops transform into matrix-vector operations in the discretization process of the 1D linear advection equation.

The following equations (1)-(4) show the process of creating matrix $x$.
For simplicity, let $n$ represent *n_element* in the program, and keep $N$ the same as that in the program.

Define column vectors $\boldsymbol{x_\ell}$ and $\boldsymbol{\xi}$ as below

$$\boldsymbol{x_\ell} = \begin{pmatrix} x_{\ell_1} \\ x_{\ell_2} \\ \vdots \\ x_{\ell_n} \end{pmatrix} = dx \begin{pmatrix} 0 \\ 1 \\ \vdots \\ n-1 \end{pmatrix} + \begin{pmatrix} dx/2 \\ dx/2 \\ \vdots \\ dx/2 \end{pmatrix} - \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \tag{1}$$

$$\boldsymbol{\xi} = \begin{pmatrix} \xi_0 \\ \xi_1 \\ \vdots \\ \xi_N \end{pmatrix} \tag{2}$$

Then the matrix $\boldsymbol{x}$ can be given by $\boldsymbol{x_\ell}$ and $\boldsymbol{\xi}$

$$\boldsymbol{x} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \boldsymbol{x_\ell}^T + (dx/2)\boldsymbol{\xi} \begin{pmatrix} 1 & 1 & \cdots & 1 \end{pmatrix} \tag{3}$$

$$= \begin{pmatrix} x_{\ell_1} & x_{\ell_2} & \cdots & x_{\ell_n} \\ x_{\ell_1} & x_{\ell_2} & \cdots & x_{\ell_n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{\ell_1} & x_{\ell_2} & \cdots & x_{\ell_n} \end{pmatrix} + (dx/2) \begin{pmatrix} \xi_0 & \xi_0 & \cdots & \xi_0 \\ \xi_1 & \xi_1 & \cdots & \xi_1 \\ \vdots & \vdots & \ddots & \vdots \\ \xi_N & \xi_N & \cdots & \xi_N \end{pmatrix} \tag{4}$$

The following equations (5)-(14) show the process of creating ODE system

$$\boldsymbol{du} = \boldsymbol{F}(\boldsymbol{u})$$

The intermediate matrix $flux\_numerical$ has the same size as matrix $\boldsymbol{du}$ and matrix $\boldsymbol{u}$. For simplicity, matrix $flux\_numerical$ is denoted as $\boldsymbol{\lambda}$ here.

By definition, matrices $\boldsymbol{du}$ and $\boldsymbol{u}$ are given as below

$$\boldsymbol{du} = \begin{pmatrix} \partial u_0^{Q_{\ell_1}}/\partial t & \partial u_0^{Q_{\ell_2}}/\partial t & \cdots & \partial u_0^{Q_{\ell_n}}/\partial t \\ \partial u_1^{Q_{\ell_1}}/\partial t & \partial u_1^{Q_{\ell_2}}/\partial t & \cdots & \partial u_1^{Q_{\ell_n}}/\partial t \\ \vdots & \vdots & \ddots & \vdots \\ \partial u_N^{Q_{\ell_1}}/\partial t & \partial u_N^{Q_{\ell_2}}/\partial t & \cdots & \partial u_N^{Q_{\ell_n}}/\partial t \end{pmatrix} \tag{5}$$

$$\boldsymbol{u} = \begin{pmatrix} u_0^{Q_{\ell_1}} & u_0^{Q_{\ell_2}} & \cdots & u_0^{Q_{\ell_n}} \\ u_1^{Q_{\ell_1}} & u_1^{Q_{\ell_2}} & \cdots & u_1^{Q_{\ell_n}} \\ \vdots & \vdots & \ddots & \vdots \\ u_N^{Q_{\ell_1}} & u_N^{Q_{\ell_2}} & \cdots & u_N^{Q_{\ell_n}} \end{pmatrix} \tag{6}$$

Define row vectors $\boldsymbol{u_0}$ and $\boldsymbol{u_N}$ as below

$$\begin{aligned} \boldsymbol{u_0} &= \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} u_0^{Q_{\ell_1}} & u_0^{Q_{\ell_2}} & \cdots & u_0^{Q_{\ell_n}} \\ u_1^{Q_{\ell_1}} & u_1^{Q_{\ell_2}} & \cdots & u_1^{Q_{\ell_n}} \\ \vdots & \vdots & \ddots & \vdots \\ u_N^{Q_{\ell_1}} & u_N^{Q_{\ell_2}} & \cdots & u_N^{Q_{\ell_n}} \end{pmatrix} \begin{pmatrix} 0 & \cdots & 0 & 1 \\ & & & 0 \\ & I_{n-1} & & \vdots \\ & & & 0 \end{pmatrix} & (7) \\ &= \begin{pmatrix} u_0^{Q_{\ell_2}} & \cdots & u_0^{Q_{\ell_n}} & u_0^{Q_{\ell_1}} \end{pmatrix} & (8) \\ \boldsymbol{u_N} &= \begin{pmatrix} 0 & \cdots & 0 & 1 \end{pmatrix} \begin{pmatrix} u_0^{Q_{\ell_1}} & u_0^{Q_{\ell_2}} & \cdots & u_0^{Q_{\ell_n}} \\ u_1^{Q_{\ell_1}} & u_1^{Q_{\ell_2}} & \cdots & u_1^{Q_{\ell_n}} \\ \vdots & \vdots & \ddots & \vdots \\ u_N^{Q_{\ell_1}} & u_N^{Q_{\ell_2}} & \cdots & u_N^{Q_{\ell_n}} \end{pmatrix} & (9) \\ &= \begin{pmatrix} u_N^{Q_{\ell_1}} & \cdots & u_N^{Q_{\ell_{n-1}}} & u_N^{Q_{\ell_n}} \end{pmatrix} & (10) \end{aligned}$$

Apply flux function pointwise to $\boldsymbol{u_0}$ and $\boldsymbol{u_N}$ and let $\boldsymbol{\gamma}$ denote the result

$$\boldsymbol{\gamma} = surface\_flux.(\boldsymbol{u_N}, \boldsymbol{u_0}, args\cdots) \tag{11}$$

$$\boldsymbol{\lambda} = \begin{pmatrix} \boldsymbol{\gamma'} \\ \boldsymbol{0} \\ \boldsymbol{\gamma} \end{pmatrix} \tag{12}$$

where $\boldsymbol{\gamma'}$ is the permutation of $\boldsymbol{\gamma}$, which is similar to the process in equation (7)

$$\boldsymbol{\gamma'} = \boldsymbol{\gamma} \begin{pmatrix} 0 & & & \\ \vdots & & I_{n-1} & \\ 0 & & & \\ 1 & 0 & \cdots & 0 \end{pmatrix} \tag{13}$$

Then the matrix $\boldsymbol{du}$ can be represented as below

$$\boldsymbol{du} = (2/dx)\left(-\boldsymbol{M}^{-1}\boldsymbol{B}\boldsymbol{\lambda} + \boldsymbol{M}^{-1}\boldsymbol{D}^{T}\boldsymbol{M}\boldsymbol{u}\right) \tag{14}$$

Thus, all the process of discretization can be completed in pure matrix-vector operations.

All the notation definitions in this demo are referenced from https://trixi-framework.github.io/Trixi.jl/stable/tutorials/scalar_linear_advection_1d/.

Huiyu Xie
April 2, 2023