UNIVERSITY OF TEXAS ARLINGTON

PHYS 5319-001: Math Methods in Physics III

Linear Algebra Package

Instructor:	Dr. Qiming Zhang
Office:	CPB 336
Phone:	817-272-2020
Email:	zhang@uta.edu

Linear Algebra Basic

- Scalar a, c

- Vector $$\boldsymbol{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

- Matrix $$\boldsymbol{A} = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}$$

Here the basis is *n*-dimension (*n* could be any positive integer).
The elements could be real or complex.

Some usual operations

- rescale a vector (for a constant *c*): *c*

$$c\boldsymbol{x} = \begin{pmatrix} cx_1 \\ \vdots \\ cx_n \end{pmatrix}$$

- addition: $\boldsymbol{x} + c\boldsymbol{y} = \boldsymbol{z}$
- sum $s = \sum_{i=1}^{n} x_i$
- find the maximum (return the index)
- scalar product $\langle \boldsymbol{x} | \boldsymbol{y} \rangle$ or $\boldsymbol{x}^T \boldsymbol{y}$
- matrix-vector $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$
- Matrix-matrix $\boldsymbol{AB} = \boldsymbol{C}$

……

$C(i,j) = 0$

loop $m = 1, n$

$C_{ij} = C_{ij} + A_{im} \cdot B_{mj}$

end

Scal

LAPACK: Linear Algebra PACKage

- Developed by netlib: https://www.netlib.org/
- LAPACK is written in Fortran 90
- Some of them are ported to python/MATLAB
- e.g. for Python `(scipy.linalg.lapack)`

for MATLAB:

Linear Algebra

Linear equations, eigenvalues, singular values, decomposition, matrix operations, matrix structure

- Including **Basic Linear Algebra Subprograms (BLAS)**

& **EISPACK**

BLAS Routines http://www.netlib.org/blas/

LEVEL 1

- **Single** *single precision*
  - SROTG - setup Givens rotation
  - SROTMG - setup modified Givens rotation
  - SROT - apply Givens rotation
  - SROTM - apply modified Givens rotation
  - SSWAP - swap x and y
  - SSCAL - x = a*x *rescal a vector*
  - SCOPY - copy x into y
  - SAXPY - y = a*x + y *vector addition*
  - SDOT - dot product
  - SDSDOT - dot product with extended precision accumulation
  - SNRM2 - Euclidean norm
  - SCNRM2- Euclidean norm
  - SASUM - sum of absolute values
  - ISAMAX - index of max abs value

LEVEL 2

- **Single**
  - SGEMV - matrix vector multiply
  - SGBMV - banded matrix vector multiply
  - SSYMV - symmetric matrix vector multiply
  - SSBMV - symmetric banded matrix vector multiply
  - SSPMV - symmetric packed matrix vector multiply
  - STRMV - triangular matrix vector multiply
  - STBMV - triangular banded matrix vector multiply
  - STPMV - triangular packed matrix vector multiply
  - STRSV - solving triangular matrix problems
  - STBSV - solving triangular banded matrix problems
  - STPSV - solving triangular packed matrix problems
  - SGER - performs the rank 1 operation A := alpha*x*y' + A
  - SSYR - performs the symmetric rank 1 operation A := alpha*x*x' + A
  - SSPR - symmetric packed rank 1 operation A := alpha*x*x' + A
  - SSYR2 - performs the symmetric rank 2 operation, A := alpha*x*y' + alpha*y*x' + A
  - SSPR2 - performs the symmetric packed rank 2 operation, A := alpha*x*y' + alpha*y*x' + A

$$\boldsymbol{y} = \alpha \boldsymbol{A}\boldsymbol{x} + \beta \boldsymbol{y}$$

LEVEL 3

- **Single**

$$C = \alpha \boldsymbol{A}\boldsymbol{B} + \beta \boldsymbol{C}$$

  - SGEMM - matrix matrix multiply
  - SSYMM - symmetric matrix matrix multiply
  - SSYRK - symmetric rank-k update to a matrix
  - SSYR2K - symmetric rank-2k update to a matrix
  - STRMM - triangular matrix matrix multiply
  - STRSM - solving triangular matrix with multiple right hand sides

- **Complex**

$$C = \alpha \boldsymbol{A}^{\dagger}\boldsymbol{B} + \beta \boldsymbol{C}$$

  - CGEMM - matrix matrix multiply
  - CSYMM - symmetric matrix matrix multiply
  - CHEMM - hermitian matrix matrix multiply
  - CSYRK - symmetric rank-k update to a matrix
  - CHERK - hermitian rank-k update to a matrix
  - CSYR2K - symmetric rank-2k update to a matrix
  - CHER2K - hermitian rank-2k update to a matrix
  - CTRMM - triangular matrix matrix multiply
  - CTRSM - solving triangular matrix with multiple right hand sides

Linear algebra equations

- *n-dim*: unknown $\{x_1, x_2, \dots, x_3\}$ satisfying

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1$$
$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2$$
$$\dots \dots$$
$$a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n$$

simply,

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \qquad m = n$$

where a_{ij} and b_i are constants, $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$.

or $\quad \boldsymbol{Ax} = \boldsymbol{b}$

Solution? Gaussian elimination

Solving linear algebra equations $\boldsymbol{Ax} = \boldsymbol{b}$

Decomposition: $\boldsymbol{A} = \boldsymbol{LU}$

$$\boldsymbol{A} = \boldsymbol{LU}$$

$$L = \begin{bmatrix} * & 0 & 0 & \cdots & 0 \\ * & * & 0 & \cdots & 0 \\ * & * & * & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ * & * & * & \cdots & * \end{bmatrix} \qquad U = \begin{bmatrix} * & * & * & \cdots & * \\ 0 & * & * & \cdots & * \\ 0 & 0 & * & \cdots & * \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & * \end{bmatrix}$$

L: lower triangular, elements $\{\alpha_{ij}\}$

U: upper triangular, elements $\{\beta_{ij}\}$

So, $\boldsymbol{Ax} = (\boldsymbol{LU})\boldsymbol{x} = \boldsymbol{L}(\boldsymbol{Ux}) = \boldsymbol{b}$ ➡ Let $\boldsymbol{Ux} \equiv \boldsymbol{y}$

Solve $\boldsymbol{Ly} = \boldsymbol{b}$ first for $\boldsymbol{y}$, then $\boldsymbol{Ux} = \boldsymbol{y}$ for $\boldsymbol{x}$, by taking advantage of Δ:

$$y_1 = b_1/\alpha_{11}, \quad y_2 = \frac{1}{\alpha_{22}}[b_2 - \alpha_{21}y_1], \ldots \qquad y_i = \frac{1}{\alpha_{ii}}\left[b_i - \sum_{j=1}^{i-1}\alpha_{ij}y_j\right]$$

$$x_n = \frac{y_n}{\beta_{nn}}, \quad \Rightarrow \quad x_i = \frac{1}{\beta_{ii}}\left[y_i - \sum_{j=i+1}^{n}\beta_{ij}x_j\right]$$

LAPACK: LU factorization for $Ax = b$

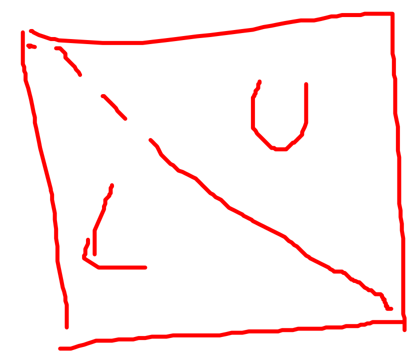
Get $A = PLU$

L:unit lower triangular

where P is an integer array, the permutation matrix

◆ dgesv()

```
subroutine dgesv ( integer                                  N,
                   integer                                  NRHS,
                   double precision, dimension( lda, * )    A,
                   integer                                  LDA,
                   integer, dimension( * )                  IPIV,
                   double precision, dimension( ldb, * )    B,
                   integer                                  LDB,
                   integer                                  INFO
                 )
```



Schrödinger Equations

$$i\hbar\frac{\partial}{\partial t}\Psi(x,t) = \left[-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V(x,t)\right]\Psi(x,t)$$

If V independent of t, separation of variables, $\Psi(x,t) = f(t)\varphi(x)$

Stationary Schrödinger equation $\hat{H}\varphi(x) = E\varphi(x)$

where $\hat{H} = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V(x)$. Bound state $E \to E_n, \{\varphi_n\}$

On the other hand,

$$\psi(x,t) = \hat{U}(t)\psi(x,0)$$

$$i\hbar\frac{\partial \hat{U}\psi(x,0)}{\partial t} = \mathcal{H}\hat{U}\psi(x,0) \quad \to \quad i\hbar\frac{\partial \hat{U}}{\partial t} = \mathcal{H}\hat{U}$$

$$\to \quad \underline{\hat{U}(t)} = e^{-i\mathcal{H}t/\hbar} \qquad \to \quad \psi(x,t) = e^{-i\hat{H}t/\hbar}\psi(x,0)$$

For any initial state $\psi(x,0) = \sum_n a_n \varphi_n(x)$,

$e^{-\hat{A}} = \sum_{n=0}^{\infty} \frac{(-)^n \hat{A}^n}{n!}$

$$\psi(x,t) = e^{-\frac{i\hat{H}t}{\hbar}}\psi(x,0) = \sum_n a_n e^{\frac{-i\hat{H}t}{\hbar}}\varphi_n(x) = \sum_n a_n e^{\frac{-iE_n t}{\hbar}}\varphi_n(x)$$

The above is still quantum mechanics. Now, just technically, let's "absorb" i in t:

$$\psi(x,\Delta t) = e^{-\frac{\hat{H}\Delta t}{\hbar}}\psi(x,0) = \sum_n a_n e^{\frac{-E_n \Delta t}{\hbar}}\varphi_n(x) = \sum_n \tilde{a}_n \varphi_n(x)$$

- If we don't now the eigenstates of H, we can start with a trial wavefunction (normalized), $\tilde{\psi}(x,0)$.
- Apply on it repeatedly, each time normalized. $\tilde{a}_0$ will grow.
- Eventually we'll get the ground state: $\tilde{\psi}(x,0) \rightarrow \varphi_0$