Two by Two Matrix Jacobians

This notebook emphasizes the multiple views of Jacobians with examples of 2x2 matrix functions.

In particular we will see the

- Symbolic "vec" format producing 4x4 matrices (generally n^2 by n^2 or mn by mn)
- Numerical formats
- The important Linear Transformation view
- Kronecker notation
- An example using ForwardDiff automatic differentiation

We also emphasize that matrix factorizations are also matrix functions, just as much as the square and the cube.

```
using Symbolics    , LinearAlgebra    , PlutoUI
```

Two by Two Matrix Jacobians


```
TableOfContents(title="Two by Two Matrix Jacobians", indent=true,aside=true)
```

Symbolic Matrices

$[p,\ q,\ r,\ s]$

```
@variables p q r s
```

X =

$$\begin{bmatrix} p & r \\ q & s \end{bmatrix}$$

```
X = [p r;q s]
```

vec

The `vec` command in Julia and in standard mathematics flattens a matrix column by column.

$[p,\ q,\ r,\ s]$

```
vec(X)
```

1) The matrix square function

```
md"""
# 1) The matrix square function
"""
```

$$\begin{bmatrix} p^2 + qr & pr + rs \\ pq + qs & qr + s^2 \end{bmatrix}$$

```
X^2
```

$[p^2 + qr,\ pq + qs,\ pr + rs,\ qr + s^2]$

```
vec(X^2)
```

Symbolic Jacobian

The Jacobian of the (flattened) matrix function X^2 symbolically

J =

$$\begin{bmatrix} 2p & r & q & 0 \\ q & p + s & 0 & q \\ r & 0 & p + s & r \\ 0 & r & q & 2s \end{bmatrix}$$

```
J = Symbolics.jacobian(vec(X^2), vec(X))
```

Numerical Jacobian

$$\begin{bmatrix} 2 & 2 & 3 & 0 \\ 3 & 5 & 0 & 3 \\ 2 & 0 & 5 & 2 \\ 0 & 2 & 3 & 8 \end{bmatrix}$$

```
begin
  M = [1 2;3 4]
  E = [.0001 .0002;.0003 .0004]
  substitute(J,Dict(p=>1,q=>3,r=>2,s=>4))
end
```

$[0.0014,\ 0.003,\ 0.002,\ 0.0044]$

```
substitute(J,Dict(p=>1,q=>3,r=>2,s=>4)) * vec(E)
```

```
2×2 Matrix{Float64}:
 0.00140007  0.0020001
 0.00300015  0.00440022
```

```
(M+E)^2 - M^2
```

Linear Transformation Jacobian

```
md"""
## Linear Transformation Jacobian
"""
```

```
linear_transformation (generic function with 1 method)
```

```
linear_transformation(E) = M*E + E*M
```

```
2×2 Matrix{Float64}:
 0.0014  0.002
 0.003   0.0044
```

```
linear_transformation(E)
```

Kronecker product or ⊗ notation

$[a,\ b,\ c,\ d]$

```
@variables a,b,c,d
```

$$\left(\begin{bmatrix} p & r \\ q & s \end{bmatrix}, \begin{bmatrix} a & c \\ b & d \end{bmatrix}\right)$$

```
[p r;q s],[a c;b d]
```

Notice all possible products with the first matrix and the second

$$\begin{bmatrix} ap & cp & ar & cr \\ bp & dp & br & dr \\ aq & cq & as & cs \\ bq & dq & bs & ds \end{bmatrix}$$

```
kron([p r;q s],[a c;b d] )
```

$$\begin{bmatrix} p & r & 0 & 0 \\ q & s & 0 & 0 \\ 0 & 0 & p & r \\ 0 & 0 & q & s \end{bmatrix}$$

```
kron(I2,X)
```

$$\begin{bmatrix} p & 0 & q & 0 \\ 0 & p & 0 & q \\ r & 0 & s & 0 \\ 0 & r & 0 & s \end{bmatrix}$$

```
kron(X',I2)
```

$$\left(\begin{bmatrix} 2p & r & q & 0 \\ q & p+s & 0 & q \\ r & 0 & p+s & r \\ 0 & r & q & 2s \end{bmatrix}, \begin{bmatrix} 2p & r & q & 0 \\ q & p+s & 0 & q \\ r & 0 & p+s & r \\ 0 & r & q & 2s \end{bmatrix} \right)$$

```
begin
    I2 = [1 0; 0 1]
    kron(I2,X) + kron(X',I2) , J
end
```

Key Kronecker identity

(A ⊗ B) * vec(C) = vec(BCA')

true

```
begin
    A = rand(2,2)
    B = rand(2,2)
    C = rand(2,2)
    kron(A,B) * vec(C) ≈ vec(B*C*A')
end
```

Useful Krockecker identities

- $$(A \otimes B)^T = A^T \otimes B^T$$
- $$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$$
- $$\det(A \otimes B) = \det(A)^m \det(B)^n$$

  , $A \in \Re^{n,n}$, $B \in \Re^{m,m}$
- $$trace(A \otimes B) = trace(A)trace(B)$$
- $$A \otimes B$$

  is orthogonal if A and B are orthogonal
- $$(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$$
- If $Au = \lambda u$, and $Bv = \mu v$, then if $X = vu^T$, then $BXA^T = \lambda\mu X$, and also $AX^TB^T = \lambda\mu X^T$. Therefore $A \otimes B$ and $B \otimes A$ have the same eigenvalues, and transposed eigenvectors.

(See **Wikipedia** for more properties.)

The Jacobian in Kronecker notation

You see (I⊗X + X'⊗I) vec(dX) = vec(XdX + dX X) = vec(d(X^2))
showing that d(X^2) = (I⊗X + X'⊗I).

Do look this over.

Automatic Differentiation (is not finite differences nor symbolic)

It comes in forward and reverse modes. Let's try forward.

```
using ForwardDiff
```

$$\begin{bmatrix} 2p & r & q & 0 \\ q & p+s & 0 & q \\ r & 0 & p+s & r \\ 0 & r & q & 2s \end{bmatrix}$$

```
J
```

```
4×4 Matrix{Int64}:
 2  2  3  0
 3  5  0  3
 2  0  5  2
 0  2  3  8
```

```
ForwardDiff.jacobian(X->X^2,M)
```

$$\begin{bmatrix} 2 & 3 & 2 & 0 \\ 2 & 5 & 0 & 2 \\ 3 & 0 & 5 & 3 \\ 0 & 3 & 2 & 8 \end{bmatrix}$$

```
#Check
substitute(J, Dict(X.=>[1 3;2 4] ))
```

2) The matrix cube Function

$$\begin{bmatrix} p^3 + qrs + 2pqr & r^2q + p^2r + s^2r + prs \\ p^2q + q^2r + s^2q + pqs & s^3 + pqr + 2qrs \end{bmatrix}$$

```
expand.(X^3)
```

Symbolic Jacobian

The Jacobian of the (flattened) matrix function X² symbolically

$$\begin{bmatrix} 3p^2 + 2qr & rs + 2pr & qs + 2pq & qr \\ qs + 2pq & p^2 + ps + s^2 + 2qr & q^2 & pq + 2qs \\ rs + 2pr & r^2 & p^2 + ps + s^2 + 2qr & pr + 2rs \\ qr & pr + 2rs & pq + 2qs & 3s^2 + 2qr \end{bmatrix}$$

```
expand.(Symbolics.jacobian(vec(X^3), vec(X)))
```

LinearTransformation Jacobian

dX X² + X dX X + dX X²

with numerical data:

```
#3 (generic function with 1 method)
```

```
E -> E*M*M + M*E*M + E*M*M
```

```
2×2 Matrix{Float64}:
 0.0111011  0.0162016
 0.0243024  0.0354035
```

```
(E+M)^3 - M^3
```

```
2×2 Matrix{Float64}:
 0.0111  0.0162
 0.0243  0.0354
```

```
(E -> E*M*M + M*E*M + E*M*M)(E)
```

check against the symbolic answer

$$\begin{bmatrix} 15 & 12 & 18 & 6 \\ 18 & 33 & 9 & 27 \\ 12 & 4 & 33 & 18 \\ 6 & 18 & 27 & 60 \end{bmatrix}$$

```
substitute( Symbolics.jacobian(vec(X^3), vec(X)) ,
Dict(p=>M[1,1],q=>M[2,1],r=>M[1,2],s=>M[2,2]))
```

$$[0.011100000000000002,\ 0.0243,\ 0.016200000000000003,\ 0.0354]$$

```
substitute( Symbolics.jacobian(vec(X^3), vec(X)) ,
Dict(p=>M[1,1],q=>M[2,1],r=>M[1,2],s=>M[2,2])) * vec(E)
```

The Jacobian in Kronecker Notation

$$\begin{bmatrix} 3p^2 + 2qr & rs + 2pr & qs + 2pq & qr \\ qs + 2pq & p^2 + ps + s^2 + 2qr & q^2 & pq + 2qs \\ rs + 2pr & r^2 & p^2 + ps + s^2 + 2qr & pr + 2rs \\ qr & pr + 2rs & pq + 2qs & 3s^2 + 2qr \end{bmatrix}$$

```
expand.( kron(I2,X^2) + kron(X',X) + kron(X'^2,I2) )
```

3) The LU Decomposition

Recall the LU Decomposition factors a matrix into unit lower-trianguar and upper triangular:

$$\left(\begin{bmatrix} 1 & 0 \\ \frac{q}{p} & 1 \end{bmatrix}, \begin{bmatrix} p & r \\ 0 & s + \frac{-qr}{p} \end{bmatrix}\right)$$

```
begin
    L,U = lu(X);
    L,U
end
```

$$\begin{bmatrix} p & r \\ q & s \end{bmatrix}$$

```
simplify_fractions.(L*U)
```

The four entries of X: p,q,r,s are transformed into these four entries in LU:

$$[\frac{q}{p},\ p,\ r,\ s + \frac{-qr}{p}]$$

```
[L[2,1],U[1,1],U[1,2],U[2,2]]
```

$$[p^2 + qr,\ pq + qs,\ pr + rs,\ qr + s^2]$$

```
vec(X^2)
```

$$\begin{bmatrix} -\frac{q}{p^2} & \frac{1}{p} & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\frac{-qr}{p^2} & \frac{-r}{p} & \frac{-q}{p} & 1 \end{bmatrix}$$

```
Symbolics.jacobian([L[2,1],U[1,1],U[1,2],U[2,2]], vec(X))
```

Exercise: Relate this to d(LU) = dL U + L dU

```
md"""
Exercise: Relate this to d(LU) = dL U + L dU
"""
```

4) Traceless symmetric eigenproblem: an example with two parameters not four

S =

$$\begin{bmatrix} p & s \\ s & -p \end{bmatrix}$$

```
S = [p s; s -p]
```

$$[\theta]$$

```
@variables θ
```

We know that the eigenvalues add to 0 (from the trace) and the eigenvectors are orthogonal (from being symmetric), so we can represent the eigenvectors and eigenvalues:

Q =

$$\begin{bmatrix} \cos\left(\frac{1}{2}\theta\right) & -\sin\left(\frac{1}{2}\theta\right) \\ \sin\left(\frac{1}{2}\theta\right) & \cos\left(\frac{1}{2}\theta\right) \end{bmatrix}$$

```
Q = [cos(θ/2) -sin(θ/2); sin(θ/2) cos(θ/2)]  # Eigenvector matrix
```

Λ =

$$\begin{bmatrix} r & 0 \\ 0 & -r \end{bmatrix}$$

```
Λ = [r 0;0 -r] # Eigenvalue matrix
```

The relationship between θ,r to p,s:

```
md"""
The relationship between θ,r to p,s:
"""
```

$$\left(\begin{bmatrix} p & s \\ s & -p \end{bmatrix}, \begin{bmatrix} r\cos(\theta) & r\sin(\theta) \\ r\sin(\theta) & r\left(-\cos^2\left(\frac{1}{2}\theta\right) + \sin^2\left(\frac{1}{2}\theta\right)\right) \end{bmatrix}, \begin{bmatrix} r\cos(\theta) & r\sin(\theta) \\ r\sin(\theta) & -r\cos(\theta) \end{bmatrix}\right)$$

```
S, simplify.(Q*Λ*Q'), [r*cos(θ) r*sin(θ) ; r*sin(θ) -r*cos(θ)]
```

Interesting mathematical observation: these are the formulas you may remember from other classes that relate cartesian coordinates to polar coordinates in the plane.

$$\left(\begin{bmatrix} \cos(\theta) & -r\sin(\theta) \\ \sin(\theta) & r\cos(\theta) \end{bmatrix}, \begin{bmatrix} \cos(\theta) & -r\sin(\theta) \\ \sin(\theta) & r\cos(\theta) \end{bmatrix}\right)$$

```
simplify.(Symbolics.jacobian( (Q*Λ*Q')[1:2] ,  [r,θ])), [cos(θ) -r*sin(θ); sin(θ)
r*cos(θ)]
```

Mathematical aside det J=r , this is the change of variables from x,y to r,θ that you may have seen in 18.02. This eigenvalue problem is the same as the cartesian coordinates to polar representations of the plane. Often written dx dy = r dr dθ

5) The full 2x2 symmetric eigenproblem

$$[\lambda_1, \ \lambda_2]$$

```
@variables λ₁ λ₂
```

We think of

$$\left(\begin{array}{cc} p & s \\ s & r \end{array} \right) = \left(\begin{array}{rr} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{array} \right) \left(\begin{array}{cc} \lambda_1 & 0 \\ 0 & \lambda_2 \end{array} \right) \left(\begin{array}{rr} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{array} \right)^T$$

as the function from

$\lambda_1,\lambda_2,\theta \rightarrow p,r,s$

```
md"""
We think of

``\left( \begin{array}{cc}
p & s \\ s & r
\end{array} \right) =
\left( \begin{array}{rr} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta)
\end{array} \right)
\left( \begin{array}{cc}
\lambda_1 & 0 \\ 0  & \lambda_2
\end{array} \right)
\left( \begin{array}{rr} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta)
\end{array} \right)^T
``
$(br)
as the function from $(br)
``\lambda_1,\lambda_2,θ  \rightarrow  p,r,s``

"""
```

S = QΛQ':

$$\begin{bmatrix} \cos^2(\theta) & \sin^2(\theta) & -2\lambda_1\cos(\theta)\sin(\theta) + 2\lambda_2\cos(\theta)\sin(\theta) \\ \sin^2(\theta) & \cos^2(\theta) & 2\lambda_1\cos(\theta)\sin(\theta) - 2\lambda_2\cos(\theta)\sin(\theta) \\ \cos(\theta)\sin(\theta) & -\cos(\theta)\sin(\theta) & \cos^2(\theta)\lambda_1 + \sin^2(\theta)\lambda_2 - \cos^2(\theta)\lambda_2 - \sin^2(\theta)\lambda_1 \end{bmatrix}$$

```
let
    Q = [cos(θ) -sin(θ); sin(θ) cos(θ)]
    S = Q*[λ₁ 0;0 λ₂]*Q'
    [p s;s r], S
    J = Symbolics.jacobian([S[1,1],S[2,2],S[1,2]] , [λ₁,λ₂,θ])
end
```

```
using Pkg
```

The determinant of this transformation simplifies to $\lambda_1 - \lambda_2$ which some people interpret as a kind of repulsion between the two eigenvalues: that is there is a tendency for the two eigenvalues to not want to be too close together. (If both are equal, when n=2, the matrix is αI, one condition takes four paremeters down to 1)