Gram Schmidt vs Modified Gram Schmidt

We have been using the classical Gram-Schmidt Algorithm. It behaves poorly under round-off error. Here is a standard example:

$$y^1 = \begin{bmatrix}1\\\varepsilon\\0\\0\end{bmatrix}, y^2 = \begin{bmatrix}1\\0\\\varepsilon\\0\end{bmatrix}, y^3 = \begin{bmatrix}1\\0\\0\\\varepsilon\end{bmatrix}, \varepsilon > 0$$

Let $\{e^1, e^2, e^3, e^4\}$ be the standard basis vectors $\left(\text{Yes, } (e^i_j) = \begin{cases}0 & i \neq j\\ 1 & i = j\end{cases}\right)$

We note that

$$y^2 = y^1 + \varepsilon(e^3 - e^2)$$
$$y^3 = y^2 + \varepsilon(e^4 - e^3)$$

and thus

$$\text{span}\{y^1, y^2\} = \text{span}\{y^1, (e^3 - e^2)\}$$
$$\text{span}\{y^1, y^2, y^3\} = \text{span}\{y^1, (e^3 - e^2), (e^4 - e^3)\}$$

Hence, GS applied to $\{y^1, y^2, y^3\}$ and $\{y^1, (e^3 - e^2), (e^4 - e^3)\}$ should produce the same orthonormal vectors. To check this, we go to MATLAB, and for $\varepsilon = 0.1$, we do indeed get the same results. You can verify this yourself. **However, with $\varepsilon = 10^{-8}$,**

$$\|Q_1 - Q_2\| = 0.5$$

where $Q_1 = [v^1, v^2, v^3]$ computed with Classical-GS for $\{y^1, y^2, y^3\}$ while $Q_2 = [v^1, v^2, v^3]$ computed with Classical-GS for $\{y^1, (e^3 - e^2), (e^4 - e^3)\}$. Hence we do NOT get the same result!

Classical Gram Schmidt Algorithm With Normalization: Initial data $\{y^1, \cdots, y^n\}$ linearly independent. Here, it is written slightly differently than in lecture:

For $k = 1 : n$
  $v^k = y^k$
  For $j = 1 : k-1$
    $v^k = v^k - \langle y^k, v^j\rangle v^j$
  End
  $v^k = \frac{v^k}{\|v^k\|}$
End

$v^1 = \frac{y^1}{\|y^1\|}$

$v^2 = y^2 - \langle y^2, v^1\rangle v^1$

$v^2 = \frac{v^2}{\|v^2\|}$

$v^3 = y^3 - \langle y^3, v^1\rangle v^1 - \langle y^3, v^2\rangle v^2$

$v^3 = v^3/\|v^3\|$, etc.

$Q_1 = [v^1, v^2, v^3]$ computed with Classical-GS for $\{y^1, y^2, y^3\}$ while $Q_2 = [v^1, v^2, v^3]$ computed with Classical-GS for $\{y^1, (e^3 - e^2), (e^4 - e^3)\}$. R_1 shows that indeed, $\{y^1, y^2, y^3\}$ is 'nearly' linearly dependent while R_2 shows that $\{y^1, (e^3-e^2), (e^4-e^3)\}$ is 'quite' linearly independent.

```
>> DemoGramSchmidtProcess
Caluclations with Classical or Standard Gram Schmidt
Epsilon = 1e-08

Q1 =

    1.0000         0         0
    0.0000   -0.7071   -0.7071
         0    0.7071         0
         0         0    0.7071


R1 =

    1.0000    1.0000    1.0000
         0    0.0000         0
         0         0    0.0000


Q2 =

    1.0000    0.0000    0.0000
    0.0000   -0.7071   -0.4082
         0    0.7071   -0.4082
         0         0    0.8165


R2 =

    1.0000   -0.0000         0
         0    1.4142   -0.7071
         0         0    1.2247

norm(Q1-Q2)

ans =

    0.5176
```

There is a modification of the Gram Schmidt Algorithm that is much better for actual calculations. You do want to know about this! For your Final Exam, you **do not** have to know the Modified-GS Algorithm itself. **All you have to know for your Final Exam is that a Modified Gram Schmidt Algorithm exists and it provides better numerical results.**

Modified Gram Schmidt

For $k = 1 : n$
    $v^k = y^k$
End
For $i = 1 : n$
    $v^i = \frac{v^i}{\|v^i\|}$
    For $j = i + 1 : n$
        $v^j = v^j - \langle v^j, v^i\rangle v^i$
    End
End

y's $\rightarrow v$'s

$v^1 = v^1 / \|v^1\|$

$j > 1, \quad \underbrace{v^j = v^j - \langle v^j, v^1\rangle v^1}$ $\therefore v^1 \perp \{v^2, \dots, v^n\} \Rightarrow v^1 \perp v^j, j \geq 2$

$v^2 = v^2 / \|v^2\|$

$j > 2 \quad \underbrace{v^j = v^j - \langle v^j, v^2\rangle v^2}$ $\quad v^2 \perp \{v^3, \dots, v^n\} \Rightarrow v^2 \perp v^j, j \geq 3$

$v^3 = v^3 / \|v^3\|$

$\vdots$

The demo code below in Canvas in the MATLAB folder

```
a=1e-8;
y1=[1 a 0 0]';
y2=[1 0 a 0]';
y3=[1 0 0 a]';

e1=[1 0 0 0]';
e2=[0 1 0 0]';
e3=[0 0 1 0]';
e4=[0 0 0 1]';

Y=[y1 y2 y3];

%Y=rand(4,4);

[Q1,R1]=GramSchmidtClassic(Y),  % Q1'*Q1-eye(3),

[Q2, R2] = GramSchmidtClassic([y1,-e2+e3,-e3+e4]),
```

```
disp('norm(Q1-Q2)')
norm(Q1-Q2)

pause


[Q3,R3]=GramSchmidtModified(Y),

[Q4,R4]=GramSchmidtModified([y1,-e2+e3,-e3+e4]),

disp('norm(Q3-Q4)')
norm(Q3-Q4)

pause


[Q5,R5]=GramSchmidtModified_MIT(Y),

[Q6,R6]=GramSchmidtModified_MIT([y1,-e2+e3,-e3+e4]),

disp('norm(Q5-Q6)')
norm(Q5-Q6)
```

$Q_3 = [v^1, v^2, v^3]$ computed with Modified-GS for $\{y^1, y^2, y^3\}$ while $Q_4 = [v^1, v^2, v^3]$ computed with Modified-GS for $\{y^1, (e^3 - e^2), (e^4 - e^3)\}$. R_3 shows that indeed, $\{y^1, y^2, y^3\}$ is 'nearly' linearly dependent while R_4 shows that $\{y^1, (e^3 - e^2), (e^4 - e^3)\}$ is 'quite' linearly independent.

```
Calculations with Modified Gram Schmidt
Epsilon = 1e-08

Q3 =

    1.0000         0         0
    0.0000   -0.7071   -0.4082
         0    0.7071   -0.4082
         0         0    0.8165


R3 =

    1.0000    1.0000    1.0000
         0    0.0000         0
         0         0    0.0000


Q4 =

    1.0000    0.0000    0.0000
    0.0000   -0.7071   -0.4082
         0    0.7071   -0.4082
         0         0    0.8165


R4 =

    1.0000   -0.0000         0
         0    1.4142   -0.7071
         0         0    1.2247

norm(Q3-Q4)

ans =

   8.1650e-09
```

Two GS Algorithms

Assume: $\{y^1, \ldots, y^n\}$ linearly independent

Classical Gram Schmidt

For $k = 1 : n$
    $v^k = y^k$
    For $j = 1 : k-1$
        $v^k = v^k - \langle y^k, v^j \rangle v^j$
    End
    $v^k = \frac{v^k}{\|v^k\|}$
End

Modified Gram Schmidt

For $k = 1 : n$
    $v^k = y^k$
End
For $i = 1 : n$
    $v^i = \frac{v^i}{\|v^i\|}$
    For $j = i+1 : n$
        $v^j = v^j - \langle v^j, v^i \rangle v^i$
    End
End

Comparison (not on any exam)

(a) Let $P_M(x)$ denote the orthogonal projection of x onto a subspace M.

(b) Classical GS: $v^1 = y^1$, and for $k \geq 2$, $v^k = y^k - P_M(y^k)$, where $M = \text{span}\{y^1, \cdots, y^{k-1}\} = \text{span}\{v^1, \cdots, v^{k-1}\}$ (optional: add in the normalization step)

(c) Modified GS:

- $v^1 = y^1$, and for $k \geq 2$, $\tilde{y}^k = y^k - P_M(y^k)$, where $M = \text{span}\{v^1\}$ (optional: add in the normalization step)
- $v^2 = \tilde{y}^2$, and for $k \geq 3$, $\tilde{y}^k = \tilde{y}^k - P_M(\tilde{y}^k)$, where $M = \text{span}\{v^2\}$ (optional: add in the normalization step)
- $v^3 = \tilde{y}^3$, and for $k \geq 4$, $\tilde{y}^k = \tilde{y}^k - P_M(\tilde{y}^k)$, where $M = \text{span}\{v^3\}$ (optional: add in the normalization step)
- etc.

You can learn more about this on the web.