Name: ____________________

LAFF Spring 15
Sample Exam 2
Spring 2014

1. Compute the following:

(a) $$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -2 & 0 \\ 0 & 1 & 0 \\ 0 & 4 & 1 \end{pmatrix} =$$

(b) $$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}^{-1} =$$

(c) $$\begin{pmatrix} 1 & -2 & 0 \\ 0 & 1 & 0 \\ 0 & 4 & 1 \end{pmatrix}^{-1} =$$

(d) $$\begin{pmatrix} 1 & -2 & 0 \\ 0 & 2 & 0 \\ 0 & 4 & 1 \end{pmatrix}^{-1} =$$

2. Assume $\delta \neq 0$.

$$\begin{pmatrix} 1 & -\alpha & 0 \\ 0 & \delta & 0 \\ 0 & -\beta & 1 \end{pmatrix}^{-1} = \begin{pmatrix} 1 & \alpha/\delta & 0 \\ 0 & 1/\delta & 0 \\ 0 & \beta/\delta & 1 \end{pmatrix}$$

Always/Sometimes/Never

Justify (prove) your answer.

3. Compute

(a) $\begin{pmatrix} -2 & 1 & 3 \end{pmatrix} \begin{pmatrix} -1 \\ 2 \\ -2 \end{pmatrix} =$

(b) $\begin{pmatrix} 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} -1 \\ 2 \\ -2 \end{pmatrix} =$

(c) $\begin{pmatrix} -2 & 1 & 3 \\ 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} -1 \\ 2 \\ -2 \end{pmatrix} =$

(d) $\begin{pmatrix} -2 & 1 & 3 \\ 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} -3 \\ 2 \\ 1 \end{pmatrix} =$

(e) $\begin{pmatrix} -2 & 1 & 3 \\ 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} -1 & -3 \\ 2 & 2 \\ -2 & 1 \end{pmatrix} =$

(f) Which of the three algorithms for computing $C := AB$ do parts (c)-(e) illustrate? (Circle the correct one.)

- Matrix-matrix multiplication by columns.
- Matrix-matrix multiplication by rows.
- Matrix-matrix multiplication via rank-1 updates.

4. Compute

(a) $2 \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix} =$

(b) $3 \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix} =$

(c) $(-1) \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix} =$

(d) $\begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix} \begin{pmatrix} 2 & 3 & -1 \end{pmatrix} =$

(e) $\begin{pmatrix} -3 \\ 0 \\ 1 \end{pmatrix} \begin{pmatrix} 1 & -2 & -1 \end{pmatrix} =$

(f) $\begin{pmatrix} 1 & -3 \\ 0 & 0 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 2 & 3 & -1 \\ 1 & -2 & -1 \end{pmatrix} =$

(g) Which of the three algorithms for computing $C := AB$ do parts (d)-(f) illustrate? (Circle the correct one.)

- Matrix-matrix multiplication by columns.
- Matrix-matrix multiplication by rows.
- Matrix-matrix multiplication via rank-1 updates.

5. Compute the LU factorization of $A = \begin{pmatrix} -2 & 1 & 0 \\ 6 & -1 & -1 \\ 4 & 2 & -1 \end{pmatrix}$ and use it to solve $Ax = b$,

where $b = \begin{pmatrix} 0 \\ 6 \\ 10 \end{pmatrix}$.

You can use any method you prefer to find the LU factorization.

6. Compute the following:

(a) $$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} =$$

(b) $$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}^{-1} \begin{pmatrix} 7 & 8 & 9 \\ 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} =$$

(c) $$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -2 & 0 & 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 1 & 0 & 0 \\ 0 & -213 & 0 & 0 & 0 & 1 & 0 \\ 0 & 512 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 1 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 & 1 & 0 & 0 \\ 0 & 213 & 0 & 0 & 0 & 1 & 0 \\ 0 & -512 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} =$$

(d) Fill in the boxes:

$$\begin{pmatrix} 1 & 0 & 0 \\ \boxed{} & 1 & 0 \\ \boxed{} & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 4 & 1 \\ 4 & 1 & 2 \\ -2 & -1 & 3 \end{pmatrix} = \begin{pmatrix} 2 & 4 & 1 \\ 0 & \boxed{} & \boxed{} \\ 0 & \boxed{} & \boxed{} \end{pmatrix}$$

7. (a) Invert $A = \begin{pmatrix} 2 & 2 & -6 \\ -4 & -5 & 14 \\ 4 & 3 & -9 \end{pmatrix}$.

(b) Does $A = \begin{pmatrix} -1 & 2 \\ 2 & -4 \end{pmatrix}$ have an inverse? Justify your answer.

8. Let A and B both be $n \times n$ matrices and both be invertible.

   $C = AB$ is invertible

   Always/Sometimes/Never

   Justify (prove) your answer.

9. Let $L_A : \mathbb{R}^3 \to \mathbb{R}^3$ be linear transformations with

$$L_A\left(\begin{pmatrix} 1 \\ 3 \\ 2 \end{pmatrix}\right) = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, L_A\left(\begin{pmatrix} -2 \\ -1 \\ 1 \end{pmatrix}\right) = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, L_A\left(\begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}\right) = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Let A be the matrix that represents linear transformation L_A. Compute

$A^{-1} =$

(Hint: it is not necessary to compute A!)

10. Let A have an inverse and let $\beta \neq 0$. Prove that $(\beta A)^{-1} = \frac{1}{\beta} A^{-1}$.

11. Evaluate

$$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}^{-1} \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -2 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{2} \end{pmatrix}^{-1} \begin{pmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix}^{-1} =$$

12. Consider the following algorithm for solving $Ux = b$, where U is upper triangular and x overwrites b.

Algorithm: $[b] := \textsc{Utrsv\_nonunit\_unb\_var2}(U, b)$

Partition $U \rightarrow \left(\begin{array}{c|c} U_{TL} & U_{TR} \\ \hline U_{BL} & U_{BR} \end{array}\right)$, $b \rightarrow \left(\begin{array}{c} b_T \\ \hline b_B \end{array}\right)$

where U_{BR} is 0×0, b_B has 0 rows

while $m(U_{BR}) < m(U)$ **do**

Repartition

$$\left(\begin{array}{c|c} U_{TL} & U_{TR} \\ \hline U_{BL} & U_{BR} \end{array}\right) \rightarrow \left(\begin{array}{cc|c} U_{00} & u_{01} & U_{02} \\ \hline u_{10}^T & \upsilon_{11} & u_{12}^T \\ \hline U_{20} & u_{21} & U_{22} \end{array}\right), \left(\begin{array}{c} b_T \\ \hline b_B \end{array}\right) \rightarrow \left(\begin{array}{c} b_0 \\ \hline \beta_1 \\ \hline b_2 \end{array}\right)$$

where υ_{11} is 1×1, β_1 has 1 row

$\beta_1 := \beta_1 / \upsilon_{11}$

$b_0 := b_0 - \beta_1 u_{01}$

Continue with

$$\left(\begin{array}{c|c} U_{TL} & U_{TR} \\ \hline U_{BL} & U_{BR} \end{array}\right) \leftarrow \left(\begin{array}{c|cc} U_{00} & u_{01} & U_{02} \\ \hline u_{10}^T & \upsilon_{11} & u_{12}^T \\ \hline U_{20} & u_{21} & U_{22} \end{array}\right), \left(\begin{array}{c} b_T \\ \hline b_B \end{array}\right) \leftarrow \left(\begin{array}{c} b_0 \\ \hline \beta_1 \\ \hline b_2 \end{array}\right)$$

endwhile

Justify that this algorithm requires approximately n^2 floating point operations.