3.1 Calculation of rotation matrix inverse

The following rotation matrix R relates two right-handed, orthogonal, unitary bases. Calculate its inverse by-hand (no calculator) in less than 30 seconds.

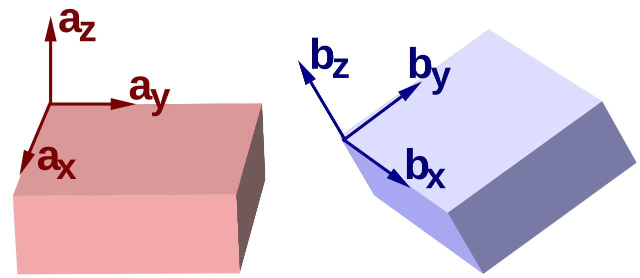
$$R = \begin{bmatrix} 0.3830 & \text{-}0.6634 & 0.6428 \\ 0.9237 & 0.2795 & \text{-}0.2620 \\ \text{-}0.0058 & 0.6941 & 0.7198 \end{bmatrix} \qquad R^{\text{-}1} = \begin{bmatrix} \boxed{} & \boxed{} & \boxed{} \\ \boxed{} & \boxed{} & \boxed{} \\ \boxed{} & \boxed{} & \boxed{} \end{bmatrix}$$

3.2 Calculating angles between unit vectors

The following rotation table ${}^{a}R^{b}$ relates right-handed, orthogonal, unit vectors $\mathbf{a}_x, \mathbf{a}_y, \mathbf{a}_z$ and $\mathbf{b}_x, \mathbf{b}_y, \mathbf{b}_z$. Calculate the angle between $\mathbf{a}_x$ and $\mathbf{b}_z$ to four (or more) significant digits.

${}^{a}R^{b}$	$\mathbf{b}_x$	$\mathbf{b}_y$	$\mathbf{b}_z$
$\mathbf{a}_x$	0.9622502	-0.08418598	0.258819
$\mathbf{a}_y$	0.1700841	0.9284017	-0.3303661
$\mathbf{a}_z$	-0.2124758	0.3619158	0.9076734

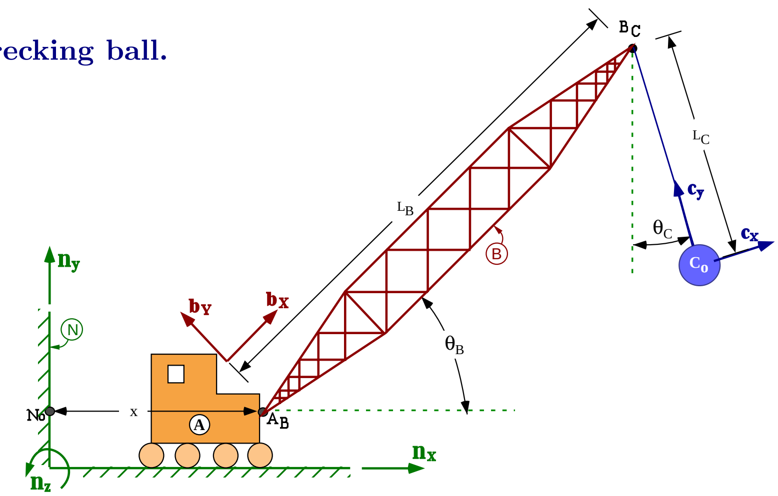
$$\angle(\mathbf{a}_x, \mathbf{b}_z) = \boxed{}^\circ$$



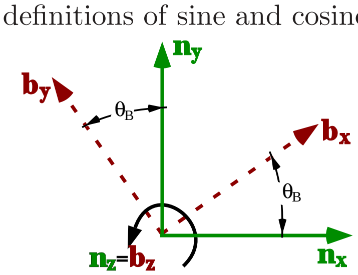
3.3 Rotation matrices for a crane and wrecking ball.

The figure to the right shows a crane whose cab A supports a boom B that swings a wrecking ball C. There are three sets of mutually perpendicular right-handed unit vectors, namely $\mathbf{n}_x$, $\mathbf{n}_y$, $\mathbf{n}_z$; $\mathbf{b}_x$, $\mathbf{b}_y$, $\mathbf{b}_z$; and $\mathbf{c}_x$, $\mathbf{c}_y$, $\mathbf{c}_z$. The point of this problem is to relate these sets of unit vectors.

Note: To relate the $\mathbf{b}_x$, $\mathbf{b}_y$, $\mathbf{b}_z$ and $\mathbf{n}_x$, $\mathbf{n}_y$, $\mathbf{n}_z$ unit vectors, it is helpful to **redraw** these vectors in a geometrically suggestive way as shown below.



(a) Use the definitions of sine and cosine to express each of $\mathbf{b}_x$, $\mathbf{b}_y$, $\mathbf{b}_z$ in terms of $\mathbf{n}_x$, $\mathbf{n}_y$, $\mathbf{n}_z$.
Result:



$$\mathbf{b}_x = \cos(\theta_B)\mathbf{n}_x + \sin(\theta_B)\mathbf{n}_y$$
$$\mathbf{b}_y = \boxed{}$$
$$\mathbf{b}_z = \boxed{}$$

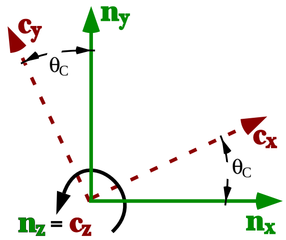
(b) Fill in the second and third rows of the ${}^{b}R^{n}$ ***rotation table*** shown to the right by extracting the various coefficients of the $\mathbf{n}_x$, $\mathbf{n}_y$, $\mathbf{n}_z$ unit vectors in the previous results.

${}^{b}R^{n}$	$\mathbf{n}_x$	$\mathbf{n}_y$	$\mathbf{n}_z$
$\mathbf{b}_x$	$\cos(\theta_B)$	$\sin(\theta_B)$	0
$\mathbf{b}_y$			
$\mathbf{b}_z$			

(c) Form ${}^{b}R^{n}$, the ***rotation matrix*** relating $\mathbf{b}_x$, $\mathbf{b}_y$, $\mathbf{b}_z$ to $\mathbf{n}_x$, $\mathbf{n}_y$, $\mathbf{n}_z$. After keeping in mind that the inverse of a rotation matrix is equal to its transpose, form the ${}^{n}R^{b}$ rotation matrix.
Result:

$$\begin{bmatrix} \mathbf{b}_x \\ \mathbf{b}_y \\ \mathbf{b}_z \end{bmatrix} = \begin{bmatrix} \quad & \quad & \quad \\ \quad & \quad & \quad \\ \quad & \quad & \quad \end{bmatrix} \begin{bmatrix} \mathbf{n}_x \\ \mathbf{n}_y \\ \mathbf{n}_z \end{bmatrix} \qquad \begin{bmatrix} \mathbf{n}_x \\ \mathbf{n}_y \\ \mathbf{n}_z \end{bmatrix} = \begin{bmatrix} \quad & \quad & \quad \\ \quad & \quad & \quad \\ \quad & \quad & \quad \end{bmatrix} \begin{bmatrix} \mathbf{b}_x \\ \mathbf{b}_y \\ \mathbf{b}_z \end{bmatrix}$$

(d) To relate the $\mathbf{c}_x$, $\mathbf{c}_y$, $\mathbf{c}_z$ and $\mathbf{n}_x$, $\mathbf{n}_y$, $\mathbf{n}_z$ unit vectors, **redraw** these vectors in a geometrically suggestive way and then use the definitions of sine and cosine to express each of $\mathbf{c}_x$, $\mathbf{c}_y$, $\mathbf{c}_z$ in terms of $\mathbf{n}_x$, $\mathbf{n}_y$, $\mathbf{n}_z$.
Result:



$$\mathbf{c}_x = \qquad$$
$$\mathbf{c}_y = \qquad$$
$$\mathbf{c}_z = \qquad$$

(e) Use the previous results to form the ${}^{c}R^{n}$ ***rotation table***.
Result:

${}^{c}R^{n}$	$\mathbf{n}_x$	$\mathbf{n}_y$	$\mathbf{n}_z$
$\mathbf{c}_x$			
$\mathbf{c}_y$			
$\mathbf{c}_z$			

(f) Use matrix multiplication to form the ${}^{b}R^{c}$ rotation table, i.e., ${}^{b}R^{c} = {}^{b}R^{n} * {}^{n}R^{c}$.
Simplify the results with the following trigonometric identities.

$$\sin(\theta_B + \theta_C) = \sin(\theta_B)\cos(\theta_C) + \sin(\theta_C)\cos(\theta_B) \qquad \cos(\text{-}\theta_C) = \cos(\theta_C)$$
$$\cos(\theta_B + \theta_C) = \cos(\theta_B)\cos(\theta_C) - \sin(\theta_B)\sin(\theta_C) \qquad \sin(\text{-}\theta_C) = \text{-}\sin(\theta_C)$$

Result:

${}^{b}R^{c}$	$\mathbf{c}_x$	$\mathbf{c}_y$	$\mathbf{c}_z$
$\mathbf{b}_x$			
$\mathbf{b}_y$			
$\mathbf{b}_z$			

(g) Verify your by-hand result by submitting the file `CraneRotationMatrices.all` which results from running the following Autolev file.

```
% File:  CraneRotationMatrices.al
%-----------------------------------
RigidFrame N, B, C
Variable   qB, qC
B.RotateZ( N, qB )   % B rotates about the "z-axis" relative to N by an angle qB
C.RotateZ( N, qC )   % C rotates about the "z-axis" relative to N by an angle qC
BC = B.GetRotationMatrix(C)  % Rotation matrix relating B and C
Save CraneRotationMatrices.all
Quit
```

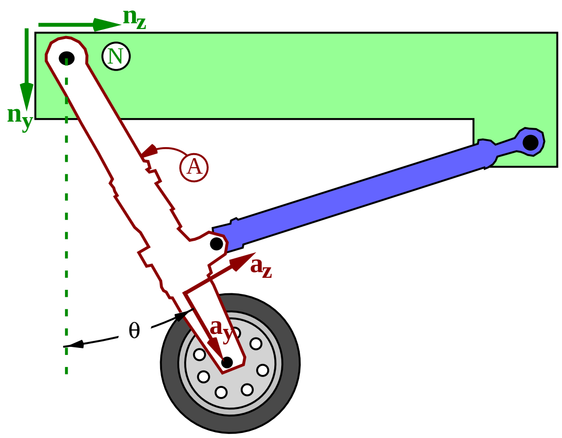
(h) Calculating rotation matrices with Autolev is **easier/harder** than doing it by hand.

3.4 Rotation tables for a landing gear system.

The figures below show three versions of the same landing gear system that consists of a strut A which has a simple rotation relative to a fuselage N. In each figure, $\mathbf{n}_x$, $\mathbf{n}_y$, $\mathbf{n}_z$ is a set of orthogonal unit vectors fixed in N and $\mathbf{a}_x$, $\mathbf{a}_y$, $\mathbf{a}_z$ is a set of orthogonal unit vectors fixed in A. However, these unit vectors have a different orientation in each figure. Determine the ${}^{a}R^{n}$ rotation table for each figure. Note, each figure has two missing vectors (e.g., $\mathbf{n}_x$ and $\mathbf{a}_x$ are missing from the first figure). Use the fact that each set of vectors is **right-handed** to add the missing vectors to each figure.

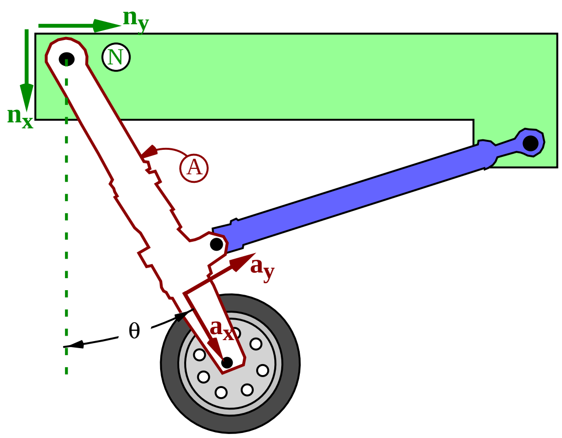
Verify your by-hand results by using Autolev's `Rotate` command. Show your work by submitting your Autolev file `SimpleRotationMatrices.al`.[1]

(a)



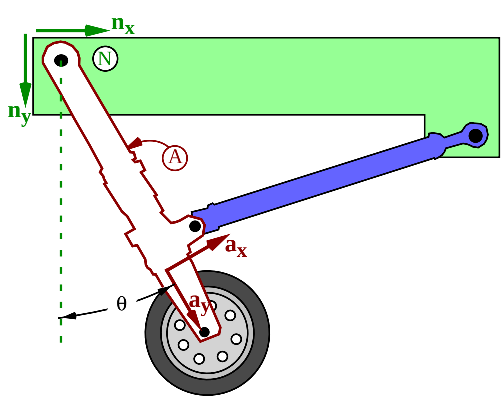
${}^{a}R^{n}$	$\mathbf{n}_x$	$\mathbf{n}_y$	$\mathbf{n}_z$
$\mathbf{a}_x$	1	0	0
$\mathbf{a}_y$	0	$\cos(\theta)$	$\sin(\theta)$
$\mathbf{a}_z$	0	$-\sin(\theta)$	$\cos(\theta)$

(b)



${}^{a}R^{n}$	$\mathbf{n}_x$	$\mathbf{n}_y$	$\mathbf{n}_z$
$\mathbf{a}_x$			
$\mathbf{a}_y$			
$\mathbf{a}_z$			

(c)



${}^{a}R^{n}$	$\mathbf{n}_x$	$\mathbf{n}_y$	$\mathbf{n}_z$
$\mathbf{a}_x$			
$\mathbf{a}_y$			
$\mathbf{a}_z$			

[1] The Autolev commands for (a) are: `RigidFrame N,A;  Variable q;  A.RotateX(N,q)`.
The Autolev commands for (c) are: `RigidFrame N,A;  Variable q;  A.RotateNegativeZ(N,q)`.