Algorithms for Manipulating Quaternions in Floating-Point Arithmetic

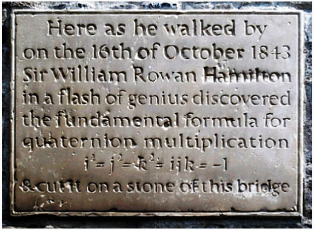
Mioara Joldeș Jean-Michel Muller

Nuscap kick-off meeting – April 2021

William Rowan Hamilton (1805–1865)

- around 1835, became fascinated by the links between $\mathbb{C}$ and 2D geometry;
- first tried to build "a 3D generalization of $\mathbb{C}$"... but cannot work with distributive and associative $\times$:
    - add a new "number" $j \notin \mathbb{C}$ and assume $\{a + ib + jc | (a, b, c) \in \mathbb{R}^3\}$ is closed under $+$ and $\times$;
    - ji must be of the form $a + ib + jc$;
    - but this gives $j = (a + ib) \times (i - c)^{-1} \in \mathbb{C}$.
- impossible even in a more general context (Frobenius 1877).

And on the 16th october of 1843...



- when walking to a meeting of the royal Irish academy in Dublin, found the solution;
- hooligan-style, carved the equations into the stone of Brougham Bridge:

$$i^2 = j^2 = k^2 = ijk = -1.$$

Quaternions

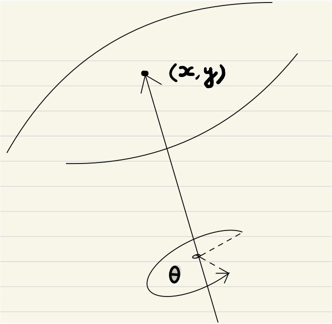
- noncommutative field $\mathbb{H}$: "numbers" of the form $q = q_0 + q_1 i + q_2 j + q_3 k$, with
  - $q_0, q_1, q_2, q_3 \in \mathbb{R}$, "components" of q;
  - i, j and k follow the (noncommutative) multiplication rules:

$\times$	1	i	j	k
1	1	i	j	k
i	i	-1	k	$-j$
j	j	$-k$	-1	i
k	k	j	$-i$	-1

  - scalar (or real) part q_0, vector part $q_1 i + q_2 j + q_3 k$.

- very trendy for a while;
- rebirth in 2nd half of 20th century, with applications in computer graphics, robotics, aerospace. . .
- TombRaider (1993) smooth 3D rotations.

Not so surprising. . .

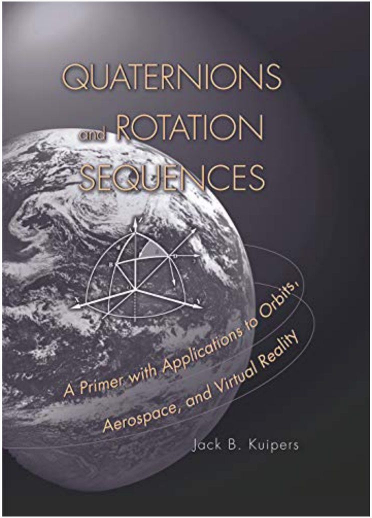


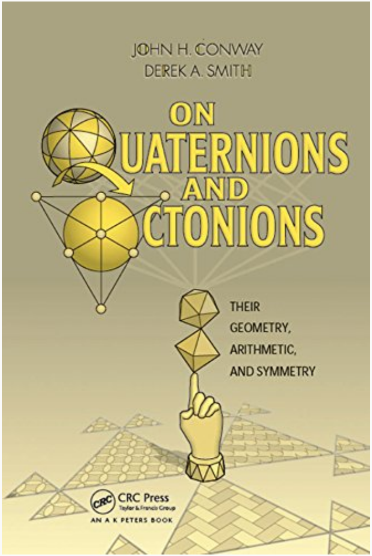
A 3-D object. . .

$$\mathcal{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}$$

. . . Represented by 9 numbers?

Two interesting references





Operations on quaternions

- scalar multiplication/division by a real number λ;
- addition of $q = q_0 + q_1 i + q_2 j + q_3 k$ and $r = r_0 + r_1 i + r_2 j + r_3 k$:

$$q + r = (q_0 + r_0) + (q_1 + r_1) \cdot i + (q_2 + r_2) \cdot j + (q_3 + r_3) \cdot j,$$

- multiplication of q and r: $q \cdot r = \pi_0 + \pi_1 i + \pi_2 j + \pi_3 k$, with

$$\begin{cases} \pi_0 &= q_0 r_0 - q_1 r_1 - q_2 r_2 - q_3 r_3, \\ \pi_1 &= q_0 r_1 + q_1 r_0 + q_2 r_3 - q_3 r_2, \\ \pi_2 &= q_0 r_2 - q_1 r_3 + q_2 r_0 + q_3 r_1, \\ \pi_3 &= q_0 r_3 + q_1 r_2 - q_2 r_1 + q_3 r_0. \end{cases}$$

Operations on quaternions

- absolute value of $q = q_0 + q_1 i + q_2 j + q_3 k$:
$$|q| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}.$$
- satisfies $|q \times q'| = |q| \cdot |q'|$ (in fact, $\times$ built for that).
- conjugate of $q = q_0 + q_1 i + q_2 j + q_3 k$:
$$\overline{q} = q_0 - q_1 i - q_2 j - q_3 k.$$
It satisfies $q\overline{q} = |q|^2$.

$\rightarrow$ reciprocal of q:
$$q^{-1} = \frac{\overline{q}}{|q|^2}.$$

- $\times$ is not commutative $\rightarrow$ no unambiguous notion of division
$\rightarrow$ avoid notation q/r (unless $q \in \mathbb{R}$), since unclear if it is $q \cdot r^{-1}$ or $r^{-1} \cdot q$.

Quaternions viewed as $\mathbb{R} \times \mathbb{R}^3$

- $q_0 + 0i + 0j + 0k$ identified with real number q_0;
- $0 + q_1 i + q_2 j + q_3 k$ identified with vector (q_1, q_2, q_3) of $\mathbb{R}^3$;
- we write $q = q_0 + v$, with $v = iq_1 + jk_2 + kq_3 = (q_1, q_2, q_3)$;
- q unit quaternion: $|q| = 1$. Gives $q = \cos\theta + u \cdot \sin\theta$, with $u = \frac{v}{|v|}$;
- If q is a quaternion, what is the function of vectors

$$w \to w' = qwq^{-1} \quad (w \in \mathbb{R}^3)$$

  - $\mathbb{R}^3 \to \mathbb{R}^3$ (one checks that real part of result is 0)
  - linear
  - $|qwq^{-1}| = |q| \cdot |w| \cdot |q^{-1}| = |w| \to$ isometry

Quaternions and 3D rotations

- $q = |q| \cdot (\cos\theta + u \cdot \sin\theta)$, rotation of angle 2θ and axis u;
- if q is a unit quaternion then $q^{-1} = \overline{q}$ so that

$$w' = qw\overline{q}.$$

- q and $-q$ represent the same rotation;
- combination of rotations $\leftrightarrow$ quaternion product:

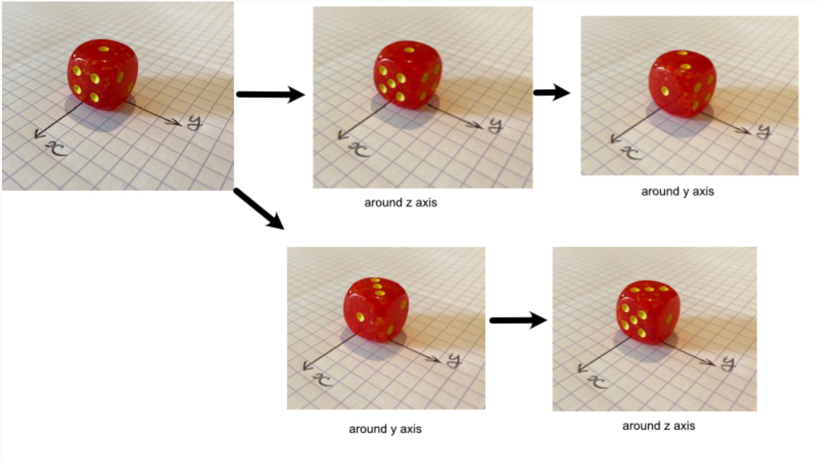
$$\begin{array}{ccl} q & \leftrightarrow & \text{rotation } \mathcal{Q} \\ r & \leftrightarrow & \text{rotation } \mathcal{R} \end{array}$$

Performing $\mathcal{Q}$ then $\mathcal{R}$ on w:

$$r\left(qwq^{-1}\right)r^{-1} = (rq)\,w\left(q^{-1}r^{-1}\right) = (rq)\,w\,(rq)^{-1}.$$

$$r \cdot q \leftrightarrow \text{rotation } \mathcal{R} \circ \mathcal{Q}.$$

Fortunately they don't commute. . .



Parameters of the underlying FP arithmetic

- underlying radix-2, precision-p FP arithmetic, extremal exponents $e_{\min}$ and $e_{\max}$;
- correctly-rounded (to nearest) FP operations, rounding function RN;
- largest finite FP number:

$$\Omega = 2^{e_{\max}+1} - 2^{e_{\max}-p+1},$$

- smallest positive nonzero number:

$$\alpha = 2^{e_{\min}-p+1},$$

- smallest positive normal number $2^{e_{\min}}$.
- rounding unit $u = 2^{-p}$.

Computing error bounds

- define $v = u/(1+u)$;
- for any t between $2^{e_{\min}}$ and Ω, we have
$$|\mathrm{RN}(t) - t| \le v \cdot |t| = \left(\frac{u}{1+u}\right) \cdot |t| < u \cdot |t|.$$
- If $\hat{q} = \hat{q}_0 + \hat{q}_1 i + \hat{q}_2 j + \hat{q}_3 k$ approximates $q = q_0 + q_1 i + q_2 j + q_3 k$, then the componentwise relative error is
$$\max_{n=0,\dots,3} \left| \frac{\hat{q}_n - q_n}{q_n} \right|,$$
(if $q_n = \hat{q}_n = 0$ then $|(\hat{q}_n - q_n)/q_n|$ is replaced by 0), and the normwise relative error is
$$\left| \frac{\hat{q} - q}{q} \right|,$$
(if $q = \hat{q} = 0$ then the normwise error is 0).

We will use several norms

- absolute value $|q| = \|q\|_2 = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}$;
- infinite norm
$$\|q\|_\infty = \max\{|q_0|, |q_1|, |q_2|, |q_3|\},$$
- 1-norm
$$\|q\|_1 = |q_0| + |q_1| + |q_2| + |q_3|.$$

They satisfy:

$$\begin{cases} \|q\|_\infty & \leq & |q| & \leq & 2 \cdot \|q\|_\infty, \\ \|q\|_\infty & \leq & \|q\|_1 & \leq & 4 \cdot \|q\|_\infty, \\ |q| & \leq & \|q\|_1 & \leq & 2 \cdot |q|. \end{cases}$$

They all have their interest

- $|\cdot| = \|\cdot\|_2$ is the natural norm of quaternions, the one that satisfies $|a \cdot b| = |a| \cdot |b|$;
- $\|\cdot\|_\infty$ is the natural norm for overflow avoidance/detection;
- $\|\cdot\|_1$ is the fastest to compute, it is computed without risk of spurious overflow/underflow;
- on my laptop (Intel Core i5 under MacOS, compiled under XCode):

$$\frac{\text{time } |\cdot|}{\text{time } \|\cdot\|_1} \approx 1.06 \qquad \frac{\text{time } \|\cdot\|_\infty}{\text{time } \|\cdot\|_1} \approx 1.76$$

Scaling a quaternion

- some libraries implement the naive formulas for $\times$, $|\cdot|$ and q^{-1};
- not a problem if input operands in a domain in which overflow & underflow are impossible or harmless (e.g., we only manipulate unit quaternions);
- otherwise: risk of spurious underflow or overflow $\rightarrow$ NaNs, infinities, or very inaccurate results.

To avoid spurious under/overflow: scaling techniques, quite similar to the ones used in complex arithmetic.

Scaling a quaternion

- $q = q_0 + q_1 i + q_2 j + q_3 k$, where q_0, q_1, q_2, and q_3 are FP numbers;
- compute a (real) scaling factor F such that
  - F is a power of 2 ($\rightarrow$ multiplication by F is errorless);
  - $\|q/F\|_\infty$ is not far from, and below, 1 (typically, will be between $1/16$ and 1).
- We can use two functions specified by the IEEE 754 Std:
  - scaleB(x, k): returns $x \cdot 2^k$ (where x is a FP number and k is an integer). Called `scalbn` in the C language;
  - logB(x): returns $\lfloor \log_2 |x| \rfloor$ (where x is a FP number).Called `logb` in C;
  - if slow, there are other solutions.

Scaling a quaternion

- natural solution: scaling factor = power of 2 immediately larger than $\max\{|q_0|, |q_1|, |q_2|, |q_3|\}$, i.e.,

$$F_\infty(q) = 2^{\lfloor \log_2 \|q\|_\infty \rfloor + 1}.$$

- on many recent architectures, $|q_0| + |q_1| + |q_2| + |q_3|$ computed more quickly than $\|q\|_\infty$ → rather use

$$F_1(q) = 2^{\lfloor \log_2 \|q\|_1 \rfloor + 1}.$$

- The definition of F_∞ implies that

$$\frac{1}{2} \leq \max_{i=1,\ldots,4} \frac{|q_i|}{F_\infty(q)} < 1,$$

From which we deduce

$$\frac{1}{8} \leq \max_{i=1,\ldots,4} \frac{|q_i|}{F_1(q)} < 1.$$

Computing the absolute value of a quaternion

- $q = q_0 + q_1 i + q_2 j + q_3 k$, where q_0, q_1, q_2, and q_3 are FP numbers
- naive algorithm:

$\hat{s}_0 \leftarrow \mathrm{RN}(q_0^2)$
$\hat{s}_1 \leftarrow \mathrm{RN}(q_1^2)$
$\hat{s}_2 \leftarrow \mathrm{RN}(q_2^2)$
$\hat{s}_3 \leftarrow \mathrm{RN}(q_3^2)$
$\hat{\sigma}_0 \leftarrow \mathrm{RN}(\hat{s}_0 + \hat{s}_1)$
$\hat{\sigma}_1 \leftarrow \mathrm{RN}(\hat{s}_2 + \hat{s}_3)$
$\hat{\sigma} \leftarrow \mathrm{RN}(\hat{\sigma}_0 + \hat{\sigma}_1)$
$\hat{N} \leftarrow \mathrm{RN}(\sqrt{\hat{\sigma}})$
return $\hat{N}$

Remarks on the naive absolute value algorithm

- spurious overflow may occur: binary32 arithmetic, $q_0 = 2^{65}$, $q_1 = q_2 = q_3 = 0$, gives $|q| = 2^{65}$ and $\hat{N} = +\infty$;
- spurious underflow may occur, but is an issue only when *all* $|q_i|$s are small (otherwise underflowing terms $\ll$ largest one). Binary32 arithmetic, $q_0 = (3/2) \times 2^{-75}$ and $q_1 = q_2 = q_3 = 0$, gives $|q| = q_0 \approx 3.97 \times 10^{-23}$, and $\hat{N} = 11863283/2^{98} \approx 3.74 \times 10^{-23}$;
- first, error analysis, assuming no under/overflow.

Analysis of the naive absolute value algorithm

1: $\hat{s}_0 \leftarrow \mathrm{RN}(q_0^2)$
2: $\hat{s}_1 \leftarrow \mathrm{RN}(q_1^2)$
3: $\hat{s}_2 \leftarrow \mathrm{RN}(q_2^2)$
4: $\hat{s}_3 \leftarrow \mathrm{RN}(q_3^2)$
5: $\hat{\sigma}_0 \leftarrow \mathrm{RN}(\hat{s}_0 + \hat{s}_1)$
6: $\hat{\sigma}_1 \leftarrow \mathrm{RN}(\hat{s}_2 + \hat{s}_3)$
7: $\hat{\sigma} \leftarrow \mathrm{RN}(\hat{\sigma}_0 + \hat{\sigma}_1)$
8: $\hat{N} \leftarrow \mathrm{RN}(\sqrt{\hat{\sigma}})$
9: **return** $\hat{N}$

$$\forall i, s_i(1-v) \le \hat{s}_i \le s_i(1+v),$$

$$\Rightarrow \forall i, \sigma_i(1-v)^2 \le \hat{\sigma}_i \le \sigma_i(1+v)^2,$$

$$\Rightarrow \sigma(1-v)^3 \le \hat{\sigma} \le \sigma(1+v)^3.$$

$$\Rightarrow \sqrt{\sigma}(1-v)^{3/2} \le \sqrt{\hat{\sigma}} \le \sqrt{\sigma}(1+v)^{3/2},$$

$$\Rightarrow N(1-v)^{5/2} \le \hat{N} = \mathrm{RN}(\sqrt{\hat{\sigma}}) \le N(1+v)^{5/2}.$$

Barring underflow or overflow, relative error bounded by $(1+v)^{5/2} - 1$, which is $< (5/2)u$.

Scaling the absolute value algorithm

- we divide q_0, q_1, q_2 and q_3 by $F = F_1(q)$ or $F_\infty(q)$ (whichever is the fastest to compute);

$\rightarrow$ new input values q'_0, q'_1, q'_2 and q'_3;

$\rightarrow$ no division: we compute $c = \textbf{logB}(\|q\|_1) + 1$ or $\textbf{logB}(\|q\|_\infty) + 1$, and

$$q'_n = \textbf{scaleB}(q_n, -c).$$

We obtain

$$\frac{1}{8} \leq \max\{|q'_0|, |q'_1|, |q'_2|, |q'_3|\} \leq 1.$$

- We apply the naive algorithm to the scaled inputs, and muliply the obtained result by F;
- Spurious overflow can no longer happen, underflow is harmless;
- Same error bound.

Computing the product of two quaternions

Naive solution: direct translation of the multiplication formula

$$\begin{cases} \hat{\pi}_0 & = & \mathrm{RN}\Big(\mathrm{RN}(\mathrm{RN}(q_0 r_0) - \mathrm{RN}(q_1 r_1)) - \mathrm{RN}(\mathrm{RN}(q_2 r_2) + \mathrm{RN}(q_3 r_3))\Big) \\ \hat{\pi}_1 & = & \mathrm{RN}\Big(\mathrm{RN}(\mathrm{RN}(q_0 r_1) + \mathrm{RN}(q_1 r_0)) + \mathrm{RN}(\mathrm{RN}(q_2 r_3) - \mathrm{RN}(q_3 r_2))\Big) \\ \hat{\pi}_2 & = & \mathrm{RN}\Big(\mathrm{RN}(\mathrm{RN}(q_0 r_2) - \mathrm{RN}(q_1 r_3)) + \mathrm{RN}(\mathrm{RN}(q_2 r_0) + \mathrm{RN}(q_3 r_1))\Big) \\ \hat{\pi}_3 & = & \mathrm{RN}\Big(\mathrm{RN}(\mathrm{RN}(q_0 r_3) + \mathrm{RN}(q_1 r_2)) - \mathrm{RN}(\mathrm{RN}(q_2 r_1) - \mathrm{RN}(q_3 r_0))\Big) \end{cases}$$

No underflow or overflow $\rightarrow$ $|\pi_n - \hat{\pi}_n| \leq u \cdot |\pi_n| + (2u + u^2) \cdot M_n.$

with

$$\begin{cases} M_0 & = & |q_0 r_0| + |q_1 r_1| + |q_2 r_2| + |q_3 r_3| \\ M_1 & = & |q_0 r_1| + |q_1 r_0| + |q_2 r_3| + |q_3 r_2| \\ M_2 & = & |q_0 r_2| + |q_1 r_3| + |q_2 r_0| + |q_3 r_1| \\ M_3 & = & |q_0 r_3| + |q_1 r_2| + |q_2 r_1| + |q_3 r_0|. \end{cases}$$

Computing the product of two quaternions

After some manipulation, gives:

$$\frac{|\pi - \hat{\pi}|}{|\pi|} \leq \sqrt{33v^2 + 72v^3 + 60v^4 + 24v^5 + 4v^6}$$

$\rightarrow$ Normwise relative error bound $\sqrt{33} \cdot u + u^2 \approx 5.75u + u^2$.

Scaling: done as for the absolute value.

Multiplication using Ogita, Rump and Oishi's's dot product

2Sum(x, y).

$$
\begin{aligned}
& s \leftarrow \mathrm{RN}(x+y) \\
& x' \leftarrow \mathrm{RN}(s-y) \\
& y' \leftarrow \mathrm{RN}(s-x') \\
& \delta_x \leftarrow \mathrm{RN}(x-x') \\
& \delta_y \leftarrow \mathrm{RN}(y-y') \\
& t \leftarrow \mathrm{RN}(\delta_x+\delta_y) \\
& \textbf{return} \;\; (s, t)
\end{aligned}
$$

Fast2Mult(x, y).

$$
\begin{aligned}
& w \leftarrow \mathrm{RN}(xy) \\
& e \leftarrow \mathrm{RN}(xy-w) \\
& \textbf{return} \;\; (w, e)
\end{aligned}
$$

Multiplication using Ogita, Rump and Oishi's's dot product

Computation of π_0:

$(s_0, e_0) \leftarrow \mathrm{Fast2Mult}(q_0, r_0)$
$(s_1, e_1) \leftarrow \mathrm{Fast2Mult}(-q_1, r_1)$
$(s_2, e_2) \leftarrow \mathrm{Fast2Mult}(-q_2, r_2)$
$(s_3, e_3) \leftarrow \mathrm{Fast2Mult}(-q_3, r_3)$
$\sigma \leftarrow e_0$
$S \leftarrow s_0$
for $i = 1$ **to** 3 **do**
    $(S, \rho) \leftarrow \mathrm{2Sum}(S, s_i)$
    $\sigma \leftarrow \mathrm{RN}(\sigma + \mathrm{RN}(\rho + e_i))$
end for
$\hat{\pi}_0 \leftarrow \mathrm{RN}(S + \sigma)$
return $\hat{\pi}_0$

When no underflow or overflow occurs, $\hat{\pi}_0$, $\hat{\pi}_1$, $\hat{\pi}_2$, and $\hat{\pi}_3$ satisfy

$$|\pi_n - \hat{\pi}_n| \leq u \cdot |\pi_n| + \frac{1}{2}\left(\frac{4u}{1-4u}\right)^2 \cdot M_n.$$

$\rightarrow$ much better bound than the naive algorithm when $M_n / |\pi_n|$ is large.

Normwise relative error bound: $u + 32u^2$.

Reciprocal computed as $\overline{q}/|q|^2$

- componentwise and normwise relative errors $\leq 4u + 5u^2 + 2u^3$;
- scaling q by $F_1(q) \rightarrow$ no overflows, harmless underflow (for the normwise error).

Example: CNES Patrius Library

File Quaternion.java

Conversion from/to rotation matrices

Rotation matrix

$$\mathcal{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}$$

Unit quaternion $q_0 + q_1 i + q_2 j + q_3 k$ associated to the same rotation.

Conversions ?

We have:

$$\mathcal{R} = 2 \times \begin{pmatrix} (q_0^2 + q_1^2) - \frac{1}{2} & q_1 q_2 - q_0 q_3 & q_1 q_3 + q_0 q_2 \\ q_1 q_2 + q_0 q_3 & (q_0^2 + q_2^2) - \frac{1}{2} & q_2 q_3 - q_0 q_1 \\ q_1 q_3 - q_0 q_2 & q_2 q_3 + q_0 q_1 & (q_0^2 + q_3^2) - \frac{1}{2} \end{pmatrix}. \qquad (1)$$

(if not unit quaternion, divide by $q_0^2 + q_1^2 + q_2^2 + q_3^2$)

Quaternion to matrix: naive implementation of (1)

- Applying (1) naively can lead to large componentwise relative error.

  Example: $q_0 = 1/2 - u$, $q_1 = 1/2 + u$, $\hat{r}_{11} = \mathrm{RN}(2 \cdot \mathrm{RN}(\mathrm{RN}(q_0^2) + \mathrm{RN}(q_1^2)) - 1)$ *gives* $\hat{r}_{11} = 0$ *whereas* $r_{11} = 4u^2$*;*

- In practice, the normwise relative error is small: we wish to show that;
- choice of matrix norm:

$$\|\mathcal{R}\|_\infty = \max_{i,j} |\mathcal{R}_{ij}|$$

- before we start, an almost ethical problem: really, what is the input?

After all, what is a unit quaternion?

- "official" answer:

$$q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1 \tag{2}$$

- Property (2) may be heavily used:
  - by the algorithm: e.g., we compute r_{22} as $2(q_0^2 + q_1^2) - 1$ instead of $1 - 2(q_1^2 + q_3^2)$ to have the same common term $2q_0^2 - 1$ everywhere in the diagonal;
  - to compute error bounds: take as example the computation of $q_0^2 + q_1^2$,
    - computation of the squares: at least one of q_0^2 and q_1^2 is $\leq 1/2$, so error $\leq u/4$ for this one, and $\leq u/2$ for the other one;
    - summing the squares: result $\leq 1 \rightarrow$ error $\leq u/2$ for the addition;
    - total error $\leq u/4 + u/2 + u/2 = 5u/4$.
- for nontrivial cases, Property (2) is never satisfied!

After all, what is a unit quaternion?

Remark 1

The only unit quaternions whose components are floating-point numbers are ± 1, $\pm i$, $\pm j$, $\pm k$ *and the numbers* $\pm \frac{1}{2} \pm \frac{1}{2} \cdot i \pm \frac{1}{2} \cdot j \pm \frac{1}{2} \cdot k$.

Lemma 1

A sum of 3 *squares of integers modulo* 8 *never equals* 7.

Proof of the Lemma: one can invoke Legendre's 3 squares theorem: *n can be written* $x^2 + y^2 + z^2$ *iff it is not of the form* $4^p(8q + 7)$.

But a simpler solution is to notice that $x^2 \mod 8 \in \{0, 1, 4\}$, and three such numbers cannot add up to 7.

After all, what is a unit quaternion?

proof of the remark:

- $q_0 + q_1 i + q_2 j + q_3 k$, with
  - $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$
  - the q_is exact in finite-precision binary arithmetic.
- Let t be the smallest integer such that all $q_i' s$ are multiple of 2^{-t}.
- We assume $t \geq 2$ (the cases 0 and 1 are processed exhaustively and lead to the cases given in the remark).
- define

  $$Q_i = 2^t q_i \in \mathbb{Z},$$

  we have

  $$Q_0^2 + Q_1^2 + Q_2^2 + Q_3^2 = 2^{2t}.$$

- t smallest $\rightarrow$ at least one of the Q_is is odd. Without l.o.g., assume it is Q_0.

After all, what is a unit quaternion?

- so far, we have

$$Q_0^2 + Q_1^2 + Q_2^2 + Q_3^2 = 2^{2t}.$$

  with Q_0 odd.
- $Q_0^2 = 1 \mod 8$
- $t \geq 2 \rightarrow 2^{2t}$ multiple of 8.
- hence, we need $Q_1^2 + Q_2^2 + Q_3^2 = 7 \mod 8$, which is impossible from the lemma.

Error of unit quaternion $\to$ rotation matrix conversion

- Reminder: we want to implement

$$\mathcal{R} = 2 \times \begin{pmatrix} (q_0^2+q_1^2)-\frac{1}{2} & q_1q_2-q_0q_3 & q_1q_3+q_0q_2 \\ q_1q_2+q_0q_3 & (q_0^2+q_2^2)-\frac{1}{2} & q_2q_3-q_0q_1 \\ q_1q_3-q_0q_2 & q_2q_3+q_0q_1 & (q_0^2+q_3^2)-\frac{1}{2} \end{pmatrix}. \qquad (1)$$

- assume $q_0^2+q_1^2+q_2^2+q_3^2=1$? It (almost) never happens;
- assume we implement transformation (1) for any input quaternion? But the transformation is meaningless for "general" quaternions, and we will get over-pessimistic results.

$\to$ assume $q_0^2+q_1^2+q_2^2+q_3^2=1+\epsilon$ for $|\epsilon|$ less than some "reasonable" bound?

Assuming $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$

- absolute error on each component $\leq 3u$;

$\rightarrow$ if $\hat{\mathcal{R}}$ is the computed value of $\mathcal{R}$,

$$\frac{\|\hat{\mathcal{R}} - \mathcal{R}\|_\infty}{\|\mathcal{R}\|_\infty} \leq \frac{3u}{\|\mathcal{R}\|_\infty}.$$

- $\mathcal{R}$ is a rotation matrix $\rightarrow$ it is orthogonal: the sum of the squares of the elements of any column in $\mathcal{R}$ is 1

$\rightarrow$ at least one element has absolute value $\geq 1/\sqrt{3}$;

$\rightarrow$ $\|\mathcal{R}\|_\infty \geq \sqrt{3}/3$.
(*lower bound on the inf. norm of a rotation matrix?*)

- Therefore, the normwise relative error is bounded by

$$3\sqrt{3}u \leq 5.197u.$$

Assuming $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1 + \epsilon$

- $\hat{\mathcal{R}}$ computed value;
- $\mathcal{R}$ exact matrix associated to the uaternion q (i.e., exact formula (1) *and* division by $q_0^2 + q_1^2 + q_2^2 + q_3^2$);
- $\mathcal{R}^*$ exact formula (1).

$$\begin{aligned}
\|\hat{\mathcal{R}} - \mathcal{R}\|_\infty &\leq \|\hat{\mathcal{R}} - \mathcal{R}^*\|_\infty + \|\mathcal{R}^* - \mathcal{R}\|_\infty \\
&\leq 3u \times 2 + |\epsilon| \cdot \|\mathcal{R}\|_\infty \\
&\leq \left(6\sqrt{3}u + |\epsilon|\right) \cdot \|\mathcal{R}\|_\infty.
\end{aligned}$$

Rotation matrix $\rightarrow$ quaternion conversion

- significantly more difficult;
- several solutions suggested. Good reference:

  S. Sarabandi and F. Thomas, A Survey on the Computation of Quaternions From Rotation Matrices. Journal of Mechanisms and Robotics, 11(2), 03 2019.

- We analyse one possible solution, employed in the Patrius Library of CNES;
- same problem as previously: what is a floating-point rotation matrix?

  In general, there's nothing such as *the exact solution.*

Rotation matrix $\rightarrow$ quaternion conversion

- remember: the rotation matrix is

$$\mathcal{R} = 2 \times \begin{pmatrix} (q_0^2 + q_1^2) - \frac{1}{2} & q_1 q_2 - q_0 q_3 & q_1 q_3 + q_0 q_2 \\ q_1 q_2 + q_0 q_3 & (q_0^2 + q_2^2) - \frac{1}{2} & q_2 q_3 - q_0 q_1 \\ q_1 q_3 - q_0 q_2 & q_2 q_3 + q_0 q_1 & (q_0^2 + q_3^2) - \frac{1}{2} \end{pmatrix}. \qquad (1)$$

- First, (1) implies

$$\begin{aligned} |q_0| &= \tfrac{1}{2}\sqrt{1 + r_{11} + r_{22} + r_{33}}, \\ |q_1| &= \tfrac{1}{2}\sqrt{1 + r_{11} - r_{22} - r_{33}}, \\ |q_2| &= \tfrac{1}{2}\sqrt{1 - r_{11} + r_{22} - r_{33}}, \\ |q_3| &= \tfrac{1}{2}\sqrt{1 - r_{11} - r_{22} + r_{33}}. \end{aligned} \qquad (3)$$

- We choose $q_0 > 0$. To be consistent q_1 has the sign of $r_{32} - r_{23}$, q_2 has the sign of $r_{13} - r_{31}$, and q_3 has the sign of $r_{21} - r_{12}$;
- Straightforward use of (3) $\rightarrow$ possible large inaccuracies if one of the terms $\pm r_{11} \pm r_{22} \pm r_{33}$ is close to -1.

Rotation matrix → quaternion conversion

Remember:

$$
\begin{aligned}
|q_0| &= \tfrac{1}{2}\sqrt{1 + r_{11} + r_{22} + r_{33}}, \\
|q_1| &= \tfrac{1}{2}\sqrt{1 + r_{11} - r_{22} - r_{33}}, \\
|q_2| &= \tfrac{1}{2}\sqrt{1 - r_{11} + r_{22} - r_{33}}, \\
|q_3| &= \tfrac{1}{2}\sqrt{1 - r_{11} - r_{22} + r_{33}}.
\end{aligned} \tag{3}
$$

- norm of the "exact" quaternion 1 → at least one of its components has absolute value $\geq 1/2$;
- For that component, the corresponding value of $\pm r_{11} \pm r_{22} \pm r_{33}$ in (3) is ≥ 0.

→ compute that component using (3), and then deduce the other components using:

$$
\begin{aligned}
4q_2q_3 &= r_{23} + r_{32}, \\
4q_1q_3 &= r_{31} + r_{13}, \\
4q_1q_2 &= r_{21} + r_{12}, \\
4q_0q_1 &= r_{32} - r_{23}, \\
4q_0q_2 &= r_{13} - r_{31}, \\
4q_0q_3 &= r_{21} - r_{12}.
\end{aligned} \tag{4}
$$

Rotation matrix → quaternion conversion

- start by successively computing the terms $\pm r_{11} \pm r_{22} \pm r_{33}$ that appear in Eq. (3) as $\mathrm{RN}(\pm r_{11} \pm \mathrm{RN}(r_{22} \pm r_{33}))$;
- As soon as we have found a term strictly $> \eta$:
  - obtain the component corresponding to that term using (3);
  - deduce the other terms using (4).
- threshold η:
  - selected based on statistical trials (Sarabandi 2019);
  - CNES Patrius Library: $\eta = -0.19$;
  - our analysis: $\eta = -2^{-e}$, for $e \in \mathbb{N}$, $0 < e < p$.

Theorem 2

When $\eta = -1/8$ and as soon as $p \geq 7$, the componentwise relative error of computing the quaternion coefficients from the rotation matrix coefficients using the method presented here is bounded by $\frac{41}{7}u + 40u^2$.

And beyond quaternions?

- there are the Octonions (Graves/Cayley) if you are ready to live with a non-associative $\times$;
- and that's it (Hurwitz theorem): the only values of n for which there exists an identity

  $$(x_1^2 + x_2^2 + \cdots + x_n^2)(y_1^2 + y_2^2 + \cdots + y_n^2) = (z_1^2 + z_2^2 + \cdots + z_n^2)$$

  where z_i is bilinear (linear both in x and y) are $1, 2, 4$, and 8.