Verified bounds for the determinant of real or complex point or interval matrices[1]

Siegfried M. Rump

Institute for Reliable Computing, Technical University of Hamburg,
Am Schwarzenberg-Campus 3, Hamburg 21073, Germany,
and Visiting Professor at Waseda University, Faculty of Science and Engineering,
3–4–1 Okubo, Shinjuku-ku, Tokyo 169–8555, Japan
*(*rump@tuhh.de*).*

Abstract

We discuss several methods to compute a verified inclusion of the determinant of a real or complex, point or interval matrix. For point matrices, large condition number 10^{15}, and large dimension ($n = 1000$) still highly accurate inclusions are computed. For real interval matrices we show that any vertex may be a unique extreme point. For wide radii we show that preconditioning may widen an inclusion significantly, and Hadamard's bound may be much better.

Key words: determinant; classical adjoint; verification methods; sharp bounds; extremely ill-conditioned matrices; NP-hard.
2010 MSC: 65G20, 15A15

1. Introduction

We discuss methods to calculate an inclusion of the determinant of a real or complex, point or interval matrix. In case of an interval matrix, an inclusion of the set of all determinants of matrices within the bounds is computed.

For the following, not much knowledge of interval arithmetic is necessary. Basically, it suffices to know that given two interval quantities $\mathbf{A}, \mathbf{B}$ and an operation $\circ$, the result $\mathbf{C}$ of the induced interval operation $\mathbf{A} \circ \mathbf{B}$ satisfies $a \circ b \in \mathbf{C}$ for all $a \in \mathbf{A}$ and all $b \in \mathbf{B}$. For more details, see [15, 12, 27, 19]. Interval quantities are in bold-face, and an expression containing an interval quantity is assumed to be evaluated using interval arithmetic.

Let $\mathbf{A} \in \mathbb{IK}^{n\times n}$ for $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$ be given. To our knowledge the first method to compute bounds for $\det(\mathbf{A}) := \{\det(A) : A \in \mathbf{A}\}$ is due to Hansen and Smith [8, Section 6]. They use a version of an LU-decomposition with partial pivoting on the midpoint matrix $\check{A}$ of $\mathbf{A}$ producing $L\check{A} \approx U$. For simplicity, they assume that the rows of $\check{A}$ were initially permuted so that L can be obtained by a numerically stable Gaussian elimination without interchanges. Then they compute $\mathbf{B} := L \cdot A$ using interval arithmetic and perform interval Gaussian elimination on $\mathbf{B}$. That results in an upper triangular interval matrix $\mathbf{T}$ with $\det(\mathbf{A}) \subseteq \prod \mathbf{T}_{ii}$.

Today we know that interval Gaussian elimination (IGA) is prone to premature failure by a pivoting element containing zero, see the analysis in [27, Section 10.1]. Moreover, it is slow because scalar rather than matrix interval operations are involved. Indeed, Table 1 shows a comparison in accuracy and computing time of the Hansen/Smith and newer methods such as in [4, 27, 17]. The table shows the median of the results for 100 random orthogonal test matrices in each dimension using INTLAB [26], the Matlab/Octave toolbox for Reliable Computing. For $n = 100$, only some 2 correct digits can be expected, from dimension $n = 120$ the method fails. It is more than an order of magnitude slower than the methods to be discussed later. Note that all matrices have condition number 1, so failure is solely due to interval dependencies.

In the following sections we introduce several methods to compute an inclusion of the determinant of point and interval matrices. We display only results for real matrices because those for complex matrices are similar, only the accuracy of the bounds is weaker by an order of magnitude.

[1]This research was partially supported by CREST, Japan Science and Technology Agency.

Table 1: Results for Hansen and Smith's method for orthogonal matrices

	Hansen/Smith			newer method	
n	failure [%]	relerr	time ratio	failure [%]	relerr
50	0	$8.0 \cdot 10^{-9}$	38.6	0	$3.9 \cdot 10^{-15}$
100	0	$5.0 \cdot 10^{-2}$	46.5	0	$8.0 \cdot 10^{-15}$
110	46	$8.4 \cdot 10^{-1}$	47.8	0	$8.8 \cdot 10^{-15}$
120	100	–	46.9	0	$9.4 \cdot 10^{-15}$

2. The determinant of a point matrix

The method of choice for computing an inclusion of $\det(A)$ seems to be preconditioning of A by one or more factors each with easily computable determinant, resulting in a matrix numerically close to the identity matrix. If the determinant of the factors is known, that reduces the problem to calculate an inclusion of $\det(I + E)$ with small $\|E\|$. Besides that standard approach, we will also introduce other methods.

For larger dimensions, several steps are necessary. First, we discuss scaling because for little larger dimensions the determinant is likely to cause over- or underflow. Next we present an accurate method to compute the determinant of a triangular matrix, followed by a new method to compute the determinant of a perturbed identity matrix. Finally, we discuss possible preconditioners eventually aiming on a highly accurate inclusion.

2.1. Scaling

For little larger dimensions the determinant may cause over- or underflow. For example, for `A = randn(n)` in Matlab notation, that is most likely the case from $n \geq 305$. More precisely, out of one million sample matrices of dimension 310, all determinants were in the over- or underflow range.

Since we are interested in larger dimensions, we henceforth assume matrices to be suitably scaled. We use the scaling factor

`f = round(sum(log2(abs(diag(lu(mid(A)))))))`

so that likely `f` $\approx \log_2(|\det(A)|)$. Then, for `q = round(f/n)` and `r = f - q*n`, we multiply A by 2^q, and then, depending on the sign of r, multiply or divide $|r|$ columns of A by 2. Mathematically, without the presence of rounding errors, the absolute value of the determinant of the resulting matrix is in the interval $[\frac{1}{\sqrt{2}}, \sqrt{2}]$. The main source of numerical errors is the approximate LU-decomposition, so that for not extremely ill-conditioned cases we can expect the scaled matrix to have determinant close to 1 in absolute value.

With respect to the value of the determinant the scaling is error-free provided that scaling itself does not cause over- or underflow. That seems rather unlikely because scaling the matrix by a factor 2 scales the determinant by 2^n, so that q and r can be expected to be small.

2.2. The determinant of a triangular matrix

For a given triangular $T \in \mathbb{K}^{n \times n}$ it seems trivial to compute an inclusion of the determinant $\det(T) = \prod T_{ii}$. Indeed, the error remains small when computed in floating-point arithmetic. However, interval operations compute a worst case error estimate, and the relative accuracy gradually decreases with the dimension. Table 2 displays in the first two columns the relative error of the computed versus true product when using floating-point and interval arithmetic, respectively.

Since the computational effort is merely $O(n)$ operations, some extra effort to increase the accuracy seems justified. Graillat [7] presented a corresponding method based on the error-free transformation TwoProduct [13].

In [14] we presented a general method to evaluate arbitrary arithmetic expressions such that under precisely specified conditions the result is faithfully rounded. Based on that general scheme we use the following algorithm to calculate an inclusion of $\prod x_i$ for $x \in \mathbb{F}^n$ with $n \geq 2$.

Table 2: Relative error of an n-fold product

n	floating-point	interval arithmetic	new pair arithmetic
50	2.2e-16	3.9e-15	8.0e-17
100	3.3e-16	8.0e-15	7.6e-17
200	4.4e-16	1.6e-14	7.9e-17
500	5.6e-16	4.0e-14	7.6e-17
1000	6.7e-16	8.0e-14	8.2e-17

```
function P = ProdAcc(x)
  n = length(x)
  [c,g] = TwoProduct(x(1),x(2));
  for i=3:n
    [c,t] = TwoProduct(c,x(i));
    g = t + x(i)*g;
  end
  G = g + midrad(0,subrealmin);
  P = c + G;
```

It has been proved in [14] that for $n \le 67,108,863$ in IEEE binary64 the floating-point sum $c + g$ is a faithfully rounded result. Note that c and g are computed in floating-point rounding to nearest, and the second last statement computes an interval `G` with left and right endpoint being the predecessor and successor of g. Hence, the computed P in the algorithm above is an inclusion of the product.

In an actual implementation extra care is taken of over- and underflow. In particular, when applying that algorithm to compute $\prod U_{ii}$ for a scaled matrix according to Subsection 2.1 for larger dimension and condition number, then it is likely that $\prod U_{ii}$ is close to 1 but `prod(diag(U))` causes intermediate over- or underflow. That suggests to devise a function `[m,e] = logdet(A)` such that $0.5 \le |m| < 1$ and $\det(A) = m \cdot 2^e$. The base 2 is chosen to avoid rounding errors, whereas $s \cdot e^{\log D} = \det(A)$ for the similar function `[logD,s] = logdet(A)` in Matlab.

The last column in Table 2 displays the relative error of the inclusion of the product computed by our compensated algorithm `ProdAcc`. As expected, the result is always of maximum accuracy.

2.3. The determinant of a perturbation of the identity matrix

Next we need to compute an inclusion of $\det(I + E)$ for a matrix $E \in \mathbb{K}^{n\times n}$ with $\|E\|$ not too large. A method of choice might be the product of the Gershgorin circles, and that argument has been used in a number of publications. However, a justification is necessary. If the Gershgorin circles $G_k(A)$ are disjoint, then clearly $\det(A) \in \prod G_k$; but for overlapping circles an individual Gershgorin circle need not contain an eigenvalue of A, and additional arguments are necessary. Indeed $\det(A) \in \prod G_k(A)$ is always true as has been shown in [25] for real, and in [5] for complex A.

But the bound by the product of Gershgorin circles can be improved. There are numerous papers on bounds of the determinant of a perturbed identity matrix, cf. [20, 21, 22, 23, 6, 11, 1, 3]. Recently we established a new bound for $\det(I + E)$ with high relative accuracy [28].

Theorem 1. *Let E be a real or complex $n \times n$ matrix and denote $\epsilon := \|E\|_F = [\mathrm{tr}(E^H E)]^{1/2}$. Suppose the spectral radius of E satisfies $\rho(E) < (1 + \epsilon^2/3)^{-1}$. Then*

$$\left| \frac{\det(I+E)}{\exp\left(\mathrm{tr}(E) - \mathrm{tr}(E^2)/2\right)} - 1 \right| \le \frac{\epsilon^2 \rho(E)}{3(1-\rho(E)) - \epsilon^2 \rho(E)} \le \frac{\epsilon^3}{3(1-\epsilon) - \epsilon^3} \,.$$

The accuracy of that bound is of order $O(\epsilon^3)$ and, because only the diagonal of E^2 is needed, it is computable in $O(n^2)$ operations. The spectral radius of E can be estimated by $\rho(E) \le \rho(|E|) \le \max_k \frac{(|E|x)_k}{x_k}$ by Perron-Frobenius Theory

and every positive $x \in \mathbb{R}^n$. Few power set iterations usually yield an accurate upper bound for $\rho(|E|)$, which in turn is a reasonable upper bound for $\rho(E)$.

The progress of Theorem 1 is the relative lower and upper bound of order $O(\epsilon^3)$, where the size of the perturbation E is limited by $1 - O(\epsilon^2)$. For ill-conditioned input matrix the perturbation E may become too large. In that case only an upper bound of $\beta := |\det(I + E)|$ can be computed. Candidates are $\rho(|E|)^n$, using Gershgorin circles, or Hadamard's bound

$$\beta \le \prod_{i=1}^{n} \|A_{i*}\|_2. \tag{1}$$

Numerical evidence suggests that the latter is (by far) the best bound.

2.4. Preconditioners

For general $A \in \mathbb{K}^{n\times n}$ the standard candidate is LU-decomposition as used in [4, 27, 17]. Consider the code:

```
[L,U,p] = lu(mid(A),'vector');
XL = inv(L);
XU = inv(U);
```

Denote $A_p := A(p,:)$. Then $A_p \approx LU$, $\det(X_L) = 1$, and $\det(X_U) = \prod(X_U)_{ii}$. By the method of computation, X_L and X_U are left inverses of L and U, respectively, see Section 14 in [9] and the picture on the cover of the book. However, as pointed out by one of the referees, it may happen that Matlab's computed approximate inverse of a triangular matrix is not triangular. An example, given by the referee, is

```
L = [1 0 0;3 2 0;2 3 1], XL = inv(L)
  L =
  1     0     0
  3     2     0
  2     3     1
XL =
  1.0000e+00   9.2519e-17  -4.4409e-17
  -1.5000e+00   5.0000e-01   2.7756e-17
  2.5000e+00  -1.5000e+00   1.0000e+00
```

This does no harm, but subsequent matrix multiplications might be more efficient if the approximate inverse is truly triangular. One remedy is to use `XL = tril(inv(L)); XU = triu(inv(U))`. However, the referee's suggestion

```
I = speye(size(A)); XL = I / L; XU = I / U;
```

seems superior. The accuracy of the residual is, sometimes, slightly worse, but the method is up to a factor 2 faster, at least in the current Matlab release 2019b. We use the sparse identity matrix which is a little faster than `eye(n)`.

Now there are several possibilities for preconditioning A. Using interval products denotes the result of any of

$$(X_U X_L)A_p \qquad X_U(X_L A_p) \qquad (XLA_p)X_U \qquad X_L(A_p X_U) \qquad (A_p X_U)XL \qquad A_p(X_U X_L) \tag{2}$$

by $\mathbf{B}$. Then $\det(A_p) \in \det(\mathbf{B})/\prod(X_U)_{ii}$. The first and third methods are used in [16], in [27] we used the fourth method. For smaller dimensions and not large condition number, there is not too much difference; however, for larger dimension $n = 1000$ and mildly ill-conditioned matrices of condition number 10^{10}, only the first two approaches succeed to compute a reasonably accurate inclusion, and of those the first one $(X_U X_L)A_p$ in (2) is the winner. In Table 3 the median relative error of the determinant inclusion for 100 samples and $n = 1000$ is shown. Matrices are randomly generated using Matlab's command `gallery('randsvd',n,cond)` generating an $n \times n$-matrix with pre-assigned singular values. This defines the first algorithm `detlu` as follows, where `detId` is based on Theorem 1.

```
function d = detlu(A)
  [L,U,p] = lu(mid(A),'vector');
  I = speye(size(A));
```

Table 3: Median relative error for the preconditioning schemes in (2) for $n = 1000$

cond(A)	$(X_U X_L)A_p$	$X_U(XLA_p)$	$(X_L A_p)X_U$	$X_L(A_p X_U)$	$(A_p X_U)X_L$	$A_p(X_U X_L)$
10	$6.8 \cdot 10^{-11}$	$1.2 \cdot 10^{-10}$	$1.1 \cdot 10^{-10}$	$1.1 \cdot 10^{-10}$	$1.2 \cdot 10^{-10}$	$6.8 \cdot 10^{-11}$
100	$1.9 \cdot 10^{-10}$	$3.7 \cdot 10^{-10}$	$3.6 \cdot 10^{-10}$	$4.3 \cdot 10^{-10}$	$4.4 \cdot 10^{-10}$	$1.9 \cdot 10^{-10}$
10^5	$3.3 \cdot 10^{-8}$	$7.7 \cdot 10^{-8}$	$7.7 \cdot 10^{-8}$	$1.2 \cdot 10^{-7}$	$1.2 \cdot 10^{-7}$	$3.3 \cdot 10^{-8}$
10^{10}	$7.0 \cdot 10^{-4}$	$2.0 \cdot 10^{-3}$	$2.2 \cdot 10^{87}$	$2.2 \cdot 10^{87}$	$1.2 \cdot 10^{296}$	$1.1 \cdot 10^{296}$

```
XL = I / L;
XU = I / U;
B = ( XU * intval(XL) ) * A(p,:);
d = parity(p)*detId(B-eye(n))/ProdAcc(diag(XU));
```

A simple and fast method in Matlab to compute the parity of the permutation vector p is `det(I(:,p))`.

One might consider an eigenvalue decomposition $AX = XD$. For $Y \approx X^{-1}$ it follows $\det(A) \in \det(M)/\det(YX)$, where $M := YAX$. One may argue that this approach works only for diagonalizable A, however, numerically, also in the presence of non-trivial Jordan blocks, M can be expected to be almost diagonal at the price of ill-conditioned X. That is because numerically eigenvectors are approximated rather than principle vectors. In turn, inclusions become wide, which is not due to the difficulty to compute the determinant but due to the method. Therefore we did not pursue this approach further.

Other preconditioning methods are the QR- and singular value decomposition. An extra difficulty is that an inclusion of the determinant of a numerically orthogonal matrix is needed. We did this using `detlu`. Both decompositions are very stable. For $A \approx QR$ we use $X_R \approx R^{-1}$, and for $A \approx U\Sigma V^*$ we use $X_\Sigma \approx \Sigma^{-1}$. Here M^* denotes the Hermitian of a matrix M. Results are shown in Table 4.

Table 4: Median relative error for other preconditioning schemes and $n = 1000$

cond(A)	$(X_U X_L)A_p$	$(Q^*A)X_R$	$Q^*(AX_R)$	AX_R	$X_\Sigma(AV^*)$	$(X_\Sigma A)V^*$	$X_\Sigma(U^*A)$	$(X_\Sigma U^*)A$
10	$6.8 \cdot 10^{-11}$	$6.6 \cdot 10^{-11}$	$6.6 \cdot 10^{-11}$	$5.8 \cdot 10^{-11}$	$2.6 \cdot 10^{-10}$	$2.6 \cdot 10^{-10}$	$1.3 \cdot 10^{-20}$	$1.3 \cdot 10^{-10}$
100	$1.9 \cdot 10^{-10}$	$7.5 \cdot 10^{-11}$	$7.6 \cdot 10^{-11}$	$6.9 \cdot 10^{-11}$	$6.3 \cdot 10^{-10}$	$6.4 \cdot 10^{-10}$	$1.3 \cdot 10^{-10}$	$1.4 \cdot 10^{-10}$
10^5	$3.2 \cdot 10^{-8}$	$4.1 \cdot 10^{-9}$	$4.6 \cdot 10^{-9}$	$4.6 \cdot 10^{-9}$	$7.8 \cdot 10^{-8}$	$7.9 \cdot 10^{-8}$	$4.4 \cdot 10^{-9}$	$4.9 \cdot 10^{-9}$
10^{10}	$7.2 \cdot 10^{-4}$	$1.6 \cdot 10^{-4}$	$1.8 \cdot 10^{-4}$	$1.8 \cdot 10^{-4}$	$1.8 \cdot 10^{-3}$	$1.8 \cdot 10^{-3}$	$1.4 \cdot 10^{-4}$	$1.6 \cdot 10^{-4}$
10^{11}	$5.6 \cdot 10^{-3}$	$1.4 \cdot 10^{-3}$	$1.6 \cdot 10^{-3}$	$1.6 \cdot 10^{-3}$	$1.6 \cdot 10^{-2}$	$1.6 \cdot 10^{-2}$	$1.2 \cdot 10^{-3}$	$1.3 \cdot 10^{-3}$

Up to now, LU-decomposition with $(X_U X_L)A_p$ and the different versions of QR-decomposition seem interesting. The last two methods seem interesting as well, however, we do not pursue them further because the computational effort for the singular value decomposition is considerably larger than for LU or QR. We mention that for an orthogonal $n \times n$ matrix Q, no method is known to decide the sign of $\det(Q)$ in $O(n^2)$ operations.

Another factorization is an approximate polar decomposition $A \approx QP$ with unitary Q and Hermitian positive definite P. Since that is usually computed via a singular value decomposition, we did not pursue that further as well.

2.5. Bounds for the determinant without preconditioning

The quality of the new bounds for the perturbed identity matrix suggests new methods to compute bounds for the determinant without preconditioning. Let `[L,U,p] = lu(A,'vector')` and abbreviate $A_p := A(p,:)$. Then it is well known from numerical analysis that $F := LU - A_p$ can be expected to be of the size of relative rounding error unit, regardless of the condition number of A. Thus $LU = A_p + F = A_p(I + A_p^{-1}F)$ and

$$\det(A) = \text{parity}(p) \cdot \prod U_{ii} / \det(I + E) \qquad \text{where} \quad E := A_p^{-1}F. \tag{3}$$

Similarly, for $F = QR - A$ it follows

$$\det(A) = \prod R_{kk} \det(Q)/\det(I + E) \qquad \text{where} \quad E := A^{-1}F. \tag{4}$$

In both cases we compute an inclusion of E using INTLAB's `verifylss`, and $\det(Q)$ is included using `detlu`. The results for $n = 1000$ and condition number up to 10^{12} are shown in Table 5. For comparison, we also show the "AX_R"-method as in Table 4. The latter, preconditioning with $X_R \approx R^{-1}$ seems simple and attractive.

Table 5: Median relative error without preconditioning schemes

	$n = 200$			$n = 1000$		
cond(A)	LU (3)	QR (4)	AX_R	LU (3)	QR (4)	AX_R
10	$3.4 \cdot 10^{-12}$	$5.4 \cdot 10^{-12}$	$5.1 \cdot 10^{-12}$	$3.7 \cdot 10^{-11}$	$5.8 \cdot 10^{-11}$	$5.8 \cdot 10^{-11}$
100	$1.2 \cdot 10^{-11}$	$1.3 \cdot 10^{-11}$	$9.7 \cdot 10^{-12}$	$1.4 \cdot 10^{-10}$	$7.0 \cdot 10^{-11}$	$6.8 \cdot 10^{-11}$
10^5	$3.2 \cdot 10^{-9}$	$4.3 \cdot 10^{-9}$	$1.7 \cdot 10^{-9}$	$2.6 \cdot 10^{-8}$	$5.7 \cdot 10^{-9}$	$4.7 \cdot 10^{-9}$
10^{10}	$1.6 \cdot 10^{-4}$	$2.3 \cdot 10^{-4}$	$6.7 \cdot 10^{-5}$	$6.0 \cdot 10^{-4}$	$2.3 \cdot 10^{-4}$	$1.8 \cdot 10^{-4}$
10^{11}	$1.5 \cdot 10^{-3}$	$2.2 \cdot 10^{-3}$	$5.9 \cdot 10^{-4}$	$4.7 \cdot 10^{-3}$	$2.0 \cdot 10^{-3}$	$1.5 \cdot 10^{-3}$
10^{12}	$1.4 \cdot 10^{-2}$	$2.1 \cdot 10^{-2}$	$5.5 \cdot 10^{-3}$	$4.2 \cdot 10^{-2}$	$1.9 \cdot 10^{-2}$	$1.4 \cdot 10^{-2}$

2.6. Bounds for the determinant with high accuracy and for ill-conditioned matrices

The bounds achieved so far give, as expected, results with accuracy depending on the condition number of the matrix. Next we use accurate dot products as in algorithm `Dot2` in [18] to produce bounds of high accuracy; the more accurate algorithm `AccSum` in [29] can be used as well. Those will also work for extremely high condition numbers beyond 10^{16}. The first method to bound the determinant is based on an LU-decomposition:

```
function d = detLU(A)
  [L,U,p] = lu(A,'vector');
  I = speye(size(A));
  XL = I / L;
  XU = I / U;
  B = Dot2(XU,XL) * A(p,:);
  d = parity(p)*detId(B-eye(n))/ProdAcc(diag(XU));
```

Here `detId(E)` calculates $\det(I + E)$ using the method described in Subsection 2.3, and `ProdAcc` is as in Subsection 2.2. The second method using accurate dot products is based on a QR-decomposition:

```
function d = detQR(A)
  XR = speye(size(A)) / triu(qr(A));
  B = Dot2(A,XR);
  d = detLU(B)/ProdAcc(diag(XR));
```

Internally, the routine `Dot2` computes an approximate term C and an interval error term E, resulting in the final inclusion $C + E$. The final improvement is to use both terms separately and use `Dot2` again on those. Here `Dot2(A,B,C)` uses the accurate dot product to compute an inclusion of $AB - C$. Note that this approach is applicable to `detQR` because the main operation is `A*XR`; it is not applicable to `detLU` because of the triple product `XU*XL*A(p,:)`.

```
function d = detQRimpr(A)
  I = speye(size(A));
  XR = I / triu(qr(A));
  [B,E] = Dot2(A,XR);       % two-fold product
  [L,U,p] = lu(B,'vector');
  XL = I / L;
```

```
XU = I / U;
[YX,F] = Dot2(XU,XL);
C = Dot2(YX,B(p,:),I) + F*B(p,:);
d = parity(p)*detId( C + (YX+F)*E(p,:) ) / ProdAcc([diag(XR);diag(XU)]);
```

The following Table 6 shows the median relative error for 100 sample matrices of sizes $n = 200$ and $n = 1000$ for different condition numbers for the three improved methods using an accurate dot product. If a relative error of the inclusion is larger than 1, then the number of such cases is printed in parenthesis in each column. In such a case still an inclusion is computed, however, it is very wide.

For condition numbers beyond 10^{14} the method based on *LU*-decomposition fails, while the *QR*-decomposition based methods succeed even for condition number 10^{16}. Up to about condition number 10^{13}, `detQRimpr` yields almost maximally accurate results.

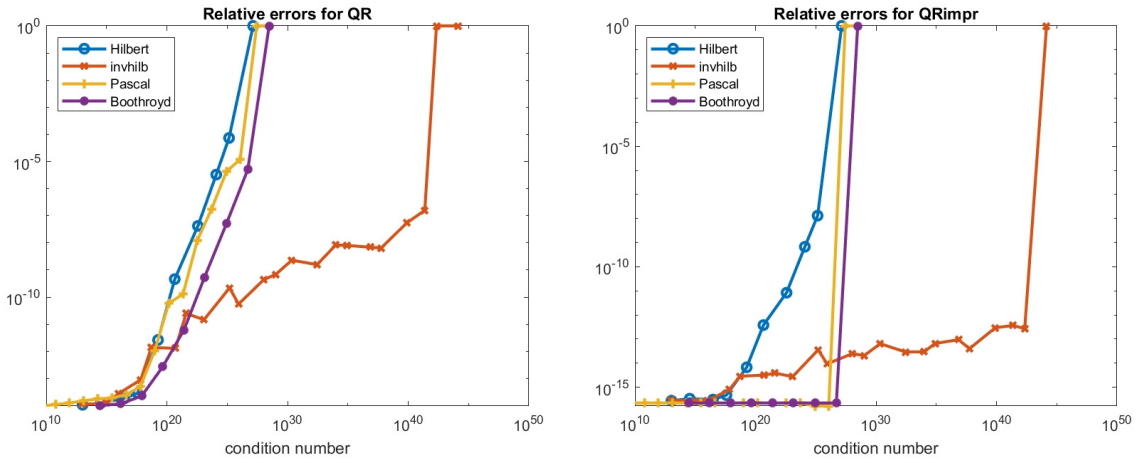
Table 6: Median relative error using accurate dot products

	$n = 200$			$n = 1000$		
cond(A)	`detLU`	`detQR`	`detQRimpr`	`detLU`	`detQR`	`detQRimpr`
10^{2}	$2.0 \cdot 10^{-12}$	$4.1 \cdot 10^{-12}$	$2.4 \cdot 10^{-16}$	$8.2 \cdot 10^{-12}$	$5.1 \cdot 10^{-11}$	$2.3 \cdot 10^{-16}$
10^{5}	$4.3 \cdot 10^{-10}$	$4.1 \cdot 10^{-12}$	$2.4 \cdot 10^{-16}$	$5.7 \cdot 10^{-10}$	$4.3 \cdot 10^{-11}$	$2.4 \cdot 10^{-16}$
10^{10}	$2.3 \cdot 10^{-5}$	$4.1 \cdot 10^{-12}$	$2.3 \cdot 10^{-16}$	$2.5 \cdot 10^{-5}$	$4.0 \cdot 10^{-11}$	$2.5 \cdot 10^{-16}$
10^{12}	$2.0 \cdot 10^{-3}$	$4.1 \cdot 10^{-12}$	$2.4 \cdot 10^{-16}$	$1.9 \cdot 10^{-3}$	$3.9 \cdot 10^{-11}$	$2.6 \cdot 10^{-16}$
10^{13}	$1.9 \cdot 10^{-2}$	$4.1 \cdot 10^{-12}$	$2.6 \cdot 10^{-16}$	$1.9 \cdot 10^{-2}$	$3.9 \cdot 10^{-11}$	$3.9 \cdot 10^{-16}$
10^{14}	$2.1 \cdot 10^{-1}$	$4.1 \cdot 10^{-12}$	$4.0 \cdot 10^{-16}$	$5.9 \cdot 10^{-1}$(39)	$2.7 \cdot 10^{-10}$	$1.7 \cdot 10^{-15}$
10^{15}	– (100)	$4.1 \cdot 10^{-12}$	$1.8 \cdot 10^{-15}$	– (100)	$1.1 \cdot 10^{-7}$	$1.3 \cdot 10^{-14}$
10^{16}	– (100)	$4.2 \cdot 10^{-12}$	$1.4 \cdot 10^{-14}$	– (100)	$5.5 \cdot 10^{-6}$(94)	$1.1 \cdot 10^{-13}$(58)

2.7. The determinant of extremely ill-conditioned matrices

The final method `detQRimpr` of the the previous subsection seems to work for extremely ill-conditioned matrices. We add a few more examples of well-known such matrices, namely, the Hilbert ('o'), inverse Hilbert ('x'), Pascal ('+') and Boothroyd ('*') matrices [2]. All matrices, possibly after integer scaling, have only integer entries. The results for the two *QR*-based methods `detQR` and `detQRimpr` of the previous subsection are shown in Figure 1. As

Figure 1: Relative error of determinant inclusion for the *QR*-based methods



can be seen, an inclusion with some accuracy succeeds up to condition number 10^{25} or so, for inverse Hilbert matrices even up to 10^{40}. The inclusions by `detQRimpr` are significantly better than those by `detQR`. Remarkably, even the simple algorithm `detQR` works well for condition number well beyond 10^{16}.

3. Interval matrices

Let $\mathbf{A} \in \mathbb{IR}^{n\times n}$ be given. We use a midpoint-radius notation $\mathbf{A} = <M,R> = \{A : M - R \le A \le M + R\}$ with inequalities to be understood componentwise. The continuity and linearity of the determinant and the expansion [10]

$$\det(A + \delta B) = \sum_{k=0}^{n} \mathrm{tr}(\mathrm{adj}_k(A)C_k(B)) \cdot \delta^k \tag{5}$$

using the k-th (classical) adjoint and compound matrix imply that there exist S_1, S_2 with $|S_\nu| = ee^T$ for $e = (1,\ldots,1)^T$ and $\det(\mathbf{A}) = [\det(M + S_1 \circ R), \det(M + S_2 \circ R)]$, where $\circ$ denotes the Hadamard (entrywise) product.

For a real interval matrix and $|S| = ee^T$ with $\det(M + S \circ R) \in \{\min\det(\mathbf{A}), \max\det(\mathbf{A})\}$, we call $M + S \circ R$ an extreme vertex of $\mathbf{A}$. Potentially 2^{2^n} such vertex determinants have to be checked. As for the componentwise distance to the nearest singular matrix, where it suffices to consider the set of 2^{2n} vertex matrices $M + S_1 ee^T S_2$ [24], one might hope to restrict the number 2^{2^n} by some strategy. The following lemma proves the contrary.

Lemma 1. *Let fixed but arbitrary $\hat{S} \in \mathbb{R}^{n\times n}$ with $|\hat{S}| = ee^T$ be given. There exists a real interval matrix $\mathbf{A} = <M,R>$ such that $M + \hat{S} \circ R$ is an extreme vertex of $\mathbf{A}$, and $\det(M + S \circ R) \ne \det(M + \hat{S} \circ R)$ for all $|S| = ee^T$ with $S \ne \hat{S}$.*

Proof. In every ε-neighborhood of $\hat{S}$ there exists a regular matrix B. For small enough ε we may choose B such that all entries of $\hat{S}$ and B have the same sign. Assume $\det(B) > 0$. Then $M := B^{-T}$ implies $\mathrm{adj}(M) = B^T/\det(B)$, and hence all entries of $\mathrm{adj}(M^T) \circ \hat{S}$ are positive. Let S with $|S| = ee^T$ and $S \ne \hat{S}$ be given. Then $\det(M + \delta S) = \det(M) + \delta \sum_{i,j}(\mathrm{adj}(M^T) \circ S)_{ij} + O(\delta^2)$ by (5), and there exists $0 < \delta \in \mathbb{R}$ such that $\det(M + \delta S) < \det(M + \delta\hat{S})$. It follows that $\det(M + \delta\hat{S}) = \max\det(\mathbf{A})$ for the unique sign matrix $\hat{S}$, and $\mathbf{A} := <M, \delta ee^T>$ proves the result. For negative $\det(B)$, we proceed similarly. □

3.1. Inclusion of the determinant for moderate radii

For small radii, an inclusion of the determinant of a real or complex interval matrix can be computed by the methods presented in the previous section, in particular `detQR` and `detQRimpr`. For not so small radii that inclusion may become wide. A powerful measure to improve an inclusion is the adjoint.

For given $\mathbf{A} \in \mathbb{IR}^{n\times n}$, define $\mathcal{S} := \mathcal{S}(\mathbf{A}) \subseteq \mathbb{R}^{n\times n}$ to be the set of all matrices S with entries

$$S_{ij} := \begin{cases} 1 & \text{if } \mathrm{adj}(\mathbf{A})_{ji} \ge 0 \\ -1 & \text{if } \mathrm{adj}(\mathbf{A})_{ji} < 0 \\ \{-1,1\} & \text{otherwise,} \end{cases} \tag{6}$$

where $\mathrm{adj}(\mathbf{A}) := \{\mathrm{adj}(A) : A \in \mathbf{A}\} \subseteq \mathbb{R}^{n\times n}$.

Theorem 2. *For given $\mathbf{A} = <M,R>$ and $\mathcal{S}$ as in (6),*

$$\min_{S\in\mathcal{S}} \det(M - S \circ R) = \min\det(\mathbf{A}) \quad \textit{and} \quad \max\det(\mathbf{A}) = \max_{S\in\mathcal{S}} \det(M + S \circ R).$$

Proof. Let $A = M + S \circ R$ with $|S| = ee^T$ be maximal, i.e., $\det(A) = \max\det(\mathbf{A})$, and let (i,j) be a fixed but arbitrary index pair. Then

$$\det(A + \varepsilon\, e_i^T e_j) = \det(A) + \varepsilon\,\mathrm{adj}(A)_{ji}.$$

If $\mathrm{adj}(A)_{ji} > 0$, the maximality of $\det(A)$ implies $S_{ij} = 1$, and similarly $S_{ij} = -1$ if $\mathrm{adj}(A)_{ji} < 0$. If $\mathrm{adj}(A)_{ji} = 0$, then S_{ij} can be replaced by 1 or -1 without changing the determinant of A. The argument for the minimum is similar. □

Suppose we want to compute an upper bound of $\det(\mathbf{A})$, where $\mathbf{B} \in \mathbb{IR}^{n\times n}$ with $\mathrm{adj}(\mathbf{A}^T) \subseteq \mathbf{B}$ is known. If $\mathbf{B}_{ij} \geq 0$ for some i, j, then, according to Theorem 2, the interval entry $\mathbf{A}_{ij}$ can be replaced by $M_{ij} + R_{ij} \in \mathbb{R}$ without changing the maximum of the determinant. Similarly, $\mathbf{A}_{ij}$ can be replaced[2] by $M_{ij} - R_{ij}$ if $\mathbf{B}_{ij} \leq 0$. Having done that for all entries $\mathbf{B}_{ij}$ not containing zero as an inner point may result in a number of point components in $\mathbf{A}$. Continuing that process with a new inclusion of $\mathrm{adj}(\mathbf{A})$, other entries of the new adjoint may not contain zero as an inner point. The same applies to the lower bound.

That approach depends on an inclusion of the adjoint of an interval matrix, which may be computed by $\mathrm{adj}(\mathbf{A}) \subseteq \det(\mathbf{A})\mathbf{A}^{-1}$. However, that sounds like a vicious because we aim for an inclusion of $\det(\mathbf{A})$. If a verification algorithm like INTLAB's `verifylss` computes an inclusion of $\mathbf{A}^{-1} := \{A^{-1} : A \in \mathbf{A}\}$, then this proves that all matrices within $\mathbf{A}$ are regular, i.e. their determinants all have the same sign. Thus we may follow the above approach two times replacing $\mathrm{adj}(\mathbf{A}^T)$ by $\mathbf{A}^{-T}$. Then taking the hull of the computed bounds delivers an inclusion of $\det(\mathbf{A})$.

The following shows INTLAB code for this refinement method:

```
function A = refine(A,sgn)
  invA = inv(A)';
  k = sum(rad(A(:)>0)); kold = inf;
  while ( k~=0 ) && ( k<kold )
    Ainf = A.inf; Asup = A.sup;
    s = ( sgn*invA(:) <= 0 );    % reduce interval entries to A.inf
    Asup(s) = Ainf(s);
    s = ( sgn*invA(:) >= 0 );    % reduce interval entries to A.sup
    Ainf(s) = Asup(s);
    A = intval(Ainf,Asup,'infsup');
    k = sum(rad(A(:))>0);
    invA = inv(A)';
  end
```

For an interval matrix $\mathbf{A}$, it follows that $\det(\mathbf{A}) = \mathrm{hull}(\det(\mathtt{A1}), \det(\mathtt{A2}))$, where `A1 = refine(A,-1)` and `A2 = refine(A,+1)`. For complex interval matrices, each entry is a circle in the complex plane. Thus, a similar approach can be defined for complex matrices, but it seems difficult to identify "vertex" matrices for the determinant. If not too

Table 7: Percentage of zero intervals of $\mathbf{A}^{-T}$ after refinement

	$n = 200$				$n = 1000$			
	% zeros intervals		# iterations		% zeros intervals		# iterations	
initial %	median	max	median	max	median	max	median	max
50	0.00	0.01	7.0	14.0	0.00	0.00	10.0	14.0
60	0.00	0.01	8.0	14.0	0.00	0.00	10.0	16.0
70	0.00	0.02	9.0	18.0	0.00	0.00	10.0	18.0
80	0.00	0.01	9.0	16.0	0.00	0.00	12.0	20.0
90	0.00	89.33	12.0	45.0	0.00	89.30	15.0	53.0

many of the entries of the inclusion of $\mathbf{A}^{-T}$ contain zero as an inner point, the method may be effective. Table 7 shows the results for 100 sample matrices with the given initial percentage of zero intervals of the inclusion of $\mathbf{A}^{-T}$. Such a matrix is constructed by choosing a random midpoint M and inflating the radius until the desired percentage of zero intervals of the inclusion of $\mathbf{A}^{-T}$ is reached. Then the median and maximal percentage of zero intervals after applying `refine` is displayed, followed by the median and maximal number of necessary iterations. For example, the value 0.01 for the maximum percentage of zero intervals for $n = 200$ indicates that, starting with 80% zero intervals, in one out of the 100 samples there was one zero interval left, otherwise none. The latter implies an accurate inclusion of $\det(\mathbf{A})$.

[2]For $\mathbf{B}_{ij} = 0$, the determinant does not depend on $\mathbf{A}_{ij}$.

If the number of entries of the inclusion of $\mathbf{A}^{-T}$ containing zero as an inner point is reduced to zero for both bounds, then the inclusion of det(**A**) is equal to the hull of the determinants of two point matrices. Consider absolute perturbations of a random matrix A resulting in an interval matrix $\mathbf{A}$ = `midrad(A,r)`. First, the radius r is successively increased by 10%, and denote by R_1 the largest value for which the relative error of `detQR(A)` is less than 1. Second, denote by R_2 the largest value of r for which `A1 = refine(A,s)` results in a point matrix for both $s = -1$ ans $s = 1$, so that a sharp inclusion of det(**A**) follows. In Table 8 we display the median and maximum of the ratios R_2/R_1 for 100 samples each. For example, for $n = 1000$ and cond$(A) = 10^4$, an inclusion with barely one correct digit is computed

Table 8: Maximal radius for absolute perturbations

	$n = 200$			$n = 1000$		
cond(A)	median(ratio)	max(ratio)	inner[%]	median(ratio)	max(ratio)	inner[%]
1	1.00	1.00	100.0	1.00	1.00	100.0
100	3.80	8.14	100.0	2.36	6.73	100.0
10^4	4.18	8.14	100.0	3.45	8.95	100.0
10^7	2.85	6.73	100.0	2.48	8.14	100.0
10^{10}	1.00	1.95	99.9	1.00	1.00	99.7

by `detQR` for a certain radius R_1, whereas using `refine` a sharp inclusion is computed in the median for a 3.45 times larger radius.

For every matrix A within $\mathbf{A} = <M, R>$, det(A) may contribute to inner bounds of det(**A**). Obvious candidates are det$(M \pm S \circ R)$ for $S = \text{sign}(M^{-T})$ obtained by the linearization of the determinant. In Table 8 we took the matrix **A** of largest radius produced by `refine` and display the median of the ratios of the diameter of the inner inclusion and of det(**A**). Since det(**A**) = hull(det$(A1)$, det$(A2)$) for those examples, the percentages reflect the true values. The inner bounds are good as long as the size of the radii cause small linearization errors.

One may argue that each iteration of `refine` costs some $O(n^3)$ operations; however, given the NP-hardness of the problem that might be acceptable.

3.2. Interval matrices with large radii

If the entries of $\mathbf{A} = <M, R>$ are so wide that an inclusion of $\mathbf{A}^{-1}$ can be computed but all components contain zero as an inner point, one may try to apply `refine` to sufficiently many minors of **A**. Another, maybe better possibility is to fix several interval entries (i, j) of **A** to its infimum or supremum. Candidates are those with $(M^{-1})_{ji}$ large in absolute value. For k such entries, however, 2^k interval determinants have to be computed. This may also be a method if an inclusion of $\mathbf{A}^{-1}$ cannot be computed.

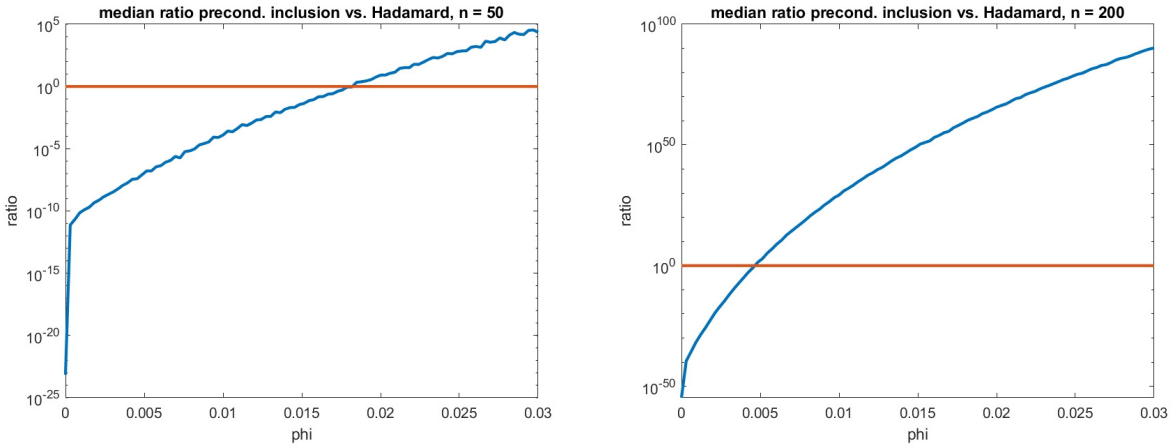
If **A** contains a matrix of rank k, then all submatrices of size $k \times k$ and larger contain a singular matrix. In that case the above approach results in many subproblems, and we do not know a method to avoid that. That reflects the NP-hardness of the problem.

In any case, preconditioning the matrix should be avoided for larger radii. For φ varying between 0 and 0.03, we generated 100 matrices by `midrad(randn(n),phi*abs(A))`, i.e., random matrices with relative radius φ. Figure 2 displays the ratio between the radius of the determinant inclusion by `detQR` and the Hadamard bound (1) for **A**. The left graph is for $n = 50$, the right one for $n = 200$. The intersection with the horizontal line at *ratio* $\equiv 1$ is the break-even point, i.e., left of it preconditioning is better, while right Hadamard's bound is superior.

Note the logarithmic scale of the ratio: for $n = 50$ and smaller radii ($\varphi \lesssim 0.018$), the inclusion by `detQR` is much better, for larger radii it becomes much worse. For $n = 200$, the break-even point is around $\varphi \lesssim 0.0047$, in both cases roughly $\varphi \lesssim 1/n$.

Acknowledgements. The author wishes to thank the anonymous referees for fruitful suggestions, particularly one of them for pointing out Matlab's behavior concerning the approximate inverse of triangular matrices.

Figure 2: Relative error of determinant inclusion for the *QR*-based methods



References

[1] T. Bhatia and T. Jain. Higher order derivatives and perturbation bounds for determinants. *Linear Algebra Appl. (LAA)*, 431:2102–2108, 2009.

[2] J. Boothroyd. Algorithm 274: Generation of Hilbert derived test matrix. *Communications of the ACM*, 9(1):11, 1966.

[3] R.P. Brent, J.H. Osborn, and W.D. Smith. Note on best possible bounds for determinants of matrices close to the identity matrix. *Linear Algebra Appl. (LAA)*, 466:21–26, 2015.

[4] H. Brönnimann, C. Burnikel, and S. Pion. Interval arithmetic yields efficient dynamic filters for computational geometry. *Discrete Applied Mathematics*, 109:25–47, 2001.

[5] F. Bünger and S.M. Rump. The determinant and complex Gershgorin circles. *Electronic Journal of Linear Algebra (ELA)*, 35:181–186, 2019.

[6] L. Elsner. Bounds for determinants of perturbed M-matrices. *Linear Algebra Appl. (LAA)*, 257:283–288, 1997.

[7] S. Graillat. Accurate floating-point product and exponentiation. *IEEE Trans. on Comp.*, 58(7):994–1000, 2009.

[8] E.R. Hansen and R. Smith. Interval arithmetic in matrix computations, Part II. *SIAM J. Numer. Anal. (SINUM)*, 4:1–9, 1967.

[9] N. J. Higham. *Accuracy and stability of numerical algorithms*. SIAM Publications, Philadelphia, 2nd edition, 2002.

[10] R.A. Horn and Ch. Johnson. *Matrix analysis*. Cambridge University Press, second edition, 2012.

[11] I.C.F. Ipsen and R. Rehman. Perturbation bounds for determinants and characteristic polynomials. *SIAM J. Matrix Anal. Appl. (SIMAX)*, 30:762–776, 2008.

[12] B. Kearfott, M. Nakao, S. Neumaier, S.M. Rump, S.P. Shary, and P. Van Hentenryck. Standardized notation in interval analysis. *Reliable Computing*, 15(1):7–13, 2010.

[13] D.E. Knuth. *The art of computer programming: Seminumerical algorithms*, volume 2. Addison Wesley, Reading, Massachusetts, 1969.

[14] M. Lange and S.M. Rump. Faithfully Rounded fLotaing-point computations. to appear in ACM TOMS.

[15] A. Neumaier. *Interval methods for systems of equations*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1990.

[16] T. Ogita. Exact determinant of integer matrices. In *4th International Workshop on Reliable Engineering Computing (REC 2010)*, pages 186–196.

[17] T. Ogita. Accurate and verified numerical computation of the matrix determinant. *Int. J. Reliability and Safety*, 6:242–254, 2012.

[18] T. Ogita, S.M. Rump, and S. Oishi. Accurate sum and dot product. *SIAM Journal on Scientific Computing (SISC)*, 26(6):1955–1988, 2005.

[19] S. Oishi, K. Ichihara, M. Kashiwagi, K. Kimura, X. Liu, H. Masai, Y. Morikura, T. Ogita, K. Ozaki, S.M. Rump, K. Sekine, A. Takayasu, and N. Yamanaka. *Principle of verified numerical computations*. Corona publisher, Tokyo, Japan, 2018 [in japanese].

[20] A. Ostrowski. Sur la détermination des bornes inférieures pour une classe des déterminants. *Bull. Sci. Math.*, 61:19–32, 1937.

[21] A.M. Ostrowski. Über die Determinanten mit überwiegender Hauptdiagonale. *Comment. Math. Helv.*, 10:69–96, 1937.

[22] A.M. Ostrowski. Sur l'approximation du déterminant de Fredholm par les déterminants des systèmes d'equations linéares. *Ark. Math. Stockholm Ser. A*, 26:1–15, 1938.

[23] G.B. Price. Bounds for determinants with dominant principal diagonal. *Proc. Amer. Math. Soc.*, 2:497–502, 1951.

[24] J. Rohn. Systems of linear interval equations. *Linear Algebra Appl. 126*, pages 39–78, 1989.

[25] S.M. Rump. Bounds for the determinant by Gershgorin circles. *Linear Algebra and its Applications (LAA)*, 563:215–219, 2019.

[26] S.M. Rump. INTLAB - INTerval LABoratory. In Tibor Csendes, editor, *Developments in Reliable Computing*, pages 77–104. Kluwer Academic Publishers, Dordrecht, 1999. `http://www.ti3.tuhh.de/intlab`.

[27] S.M. Rump. Verification methods: Rigorous results using floating-point arithmetic. *Acta Numerica*, 19:287–449, 2010.

[28] S.M. Rump and P. Batra. Addendum to the determinant of a perturbed identity matrix. *Linear Algebra and its Applications (LAA)*, 565:309–312, 2019.

[29] S.M. Rump, T. Ogita, and S. Oishi. Accurate floating-point summation part I: Faithful rounding. *SIAM J. Sci. Comput.*, 31(1):189–224, 2008.