
# On the Precision Required in SDP Pipelines

Document Number ..... SDP Memo 32
Document Type ..... 032
Revision ..... 2
Author ..... Stefano Salvini
Release Date ..... 2018-07-03
Document Classification ..... Unrestricted
Status ..... Advanced Draft

| Lead Author | Designation | Affiliation |
|---|---|---|
| Stefano Salvini | | University of Oxford |
| Signature & Date:  Stefano Salvini (Nov 1, 2018) | | |

| Owned by | Designation | Affiliation |
|---|---|---|
| | | |
| Signature & Date: | | |

| Approved by | Designation | Affiliation |
|---|---|---|
| | | |
| Signature & Date: | | |

| Released by | Designation | Affiliation |
|---|---|---|
| | | |
| Signature & Date: | | |

| **Revision** | **Date of issue** | **Prepared by** | **Comments** |
|---|---|---|---|
| 2 | 2018-07-03 | Stefano Salvini | Advanced draft |

**ORGANISATION DETAILS**

| | |
|---|---|
| Name | Science Data Processor Consortium |
| Address | Astrophysics<br>Cavendish Laboratory<br>JJ Thomson Avenue<br>Cambridge CB3 0HE |
| Website | http://ska-sdp.org |
| Email | ska-sdp-pa@mrao.cam.ac.uk |

# Table of Contents

## Summary

The issue of the format of numerical data within SDP pipelines is a very practical one. Costs increase by about a factor two when using double precision (64 bits) rather than single precision floating point numbers (32 bits): storage needs double, fewer items can be kept in fast memory (on-chip and otherwise), computational speed halves. So, the numerical precision used has a direct effect on SKA costs and needs to be tackled.

This documents employs a numerical analysis approach to the issue of the numerical precision required in the SDP pipelines, taking into account the observed data errors, the number of samples or observations, the operations carried out during the reductions, and the error properties of these functions. Necessarily, this provides qualitative, possibly semi-quantitative, results.

The arguments presented here lead to a strategy for precision handling, making sure that no digits are lost unnecessarily by switching to higher precision only when required.

While this paper concentrates on the imaging pipeline, its arguments could be easily extended to other SKA pipelines. It should also be noticed that the same arguments apply to both unpolarized and fully polarized situations. This particular point, as it is considered obvious from the point of view of this paper, will not be laboured further.

# 1 Symbols, their meaning and General Considerations

Table 1 below lists the symbols used in this document.

In this document the word *receiver* is used to denote a dish for MID, a station (aperture arrays) for LOW. In other words the receiving entities whose measurements are cross-correlated.

| Symbol | Description |
|---|---|
| $N_A$ | Number of Antennas within a receiver (stations):<br>$N_A > 1$, for stations (LOW)<br>$N_A = 1$, for dishes (MID) |
| $N_R$ | Number of receivers (Antennas for MID, stations for LOW (AA)) |
| $B_R$ | Number of bits from receiver |
| $D$ | Required dynamic range, factor (e.g. $10^3$) |
| $\Xi$ | Signal to Noise Ratio on the antennas (power) (e.g.$10^{-3}$) |
| $S$ | Sampling rate (e.g. 600 MHz) |
| $N_C$ | Number of channels (frequencies) |
| $T_D$ | Cross-correlation dump-time |
| $T_C$ | Dump-time for calibration: an integer multiple of $T_D$ |
| $T_O$ | Total Observation time in seconds |
| $H_O$ | Total Observation time in hours |
| $X_2$ | Number of significant binary digits (bits) of $X$ |
| $X_{10}$ | Number of significant decimal digits of $X$; $X_{10} = X_2 \log_{10} 2$ |
| $L_G$ | Binary digits lost by well conditioned operations, such as FFT, product and sums of vectors, etc. In general, these are only very modest values.<br>Please, see Section 2. |

**Table 1:** Symbols used in the document

## 1.1 Why analysing the required number precision by numerical analysis?

In many cases, analysis have been carried out based on specific examples. This sort of *anecdotical* studies can only have limited significance as they apply to specific examples and, possibly, to very specific situations. Moreover, error estimates are very difficult to obtain, if at all possible, and posit a number of assumptions, some based on experience, others, possibly, on (oral?) tradition.

To my knowledge, no sensitivity analysis studies on these have been carried out to date. Those would require statistical analysis of the results obtained when modifying the input. So, knowledge of the error bounds is episodical, and cannot be easily extrapolated to other cases. Of course, computing everything using double precision numbers would solve the problem. This should be traded against increasing computational costs: double precision computation is about a factor two slower than single precision; twice the memory and storage space would be necessary; the speed for the same number of data transfers would halve.

Cost containment, essential to build SKA, would require the use of lower precision (single precision) until a switch to higher precision became necessary to avoid losing information. Studying the accumulation and loss of *significant digits*, i.e. avoiding the growth of error, from observation to imaging is a valid strategy to tackle the issue.

It must be stressed that errors and significant digits must be viewed in this document as *norm-wise*, i.e. relating to the norms of the errors in the image, etc., to be seen as statistically significant but certainly not applicable to individual pixels.

Issues such as RFI and the excision of incorrect data are not dealt with here. In any case, these have little to contribute to the significant digits propagation throughout the pipeline.

Likewise, details of the antennas are ignored altogether: however, the impact of the SNR on each of them is significant in terms of convergence hence the number of digits acquired. In the case of LOW, obviously antennas are ”averaged”, i.e. summed within a station and that contribute to the acquisition of significant digits.

## 1.2 Absolute vs Relative errors

The distinction between absolute an relative errors (hence number of digits available) is essential here. Given two quantities $x$ and $y$, for example vectors, the absolute error $\varepsilon_a$ is given by the norm of their difference:

$$\varepsilon_A = \|x - y\|$$

. The relative error $\varepsilon_R$ is given by normalising (normwise) the error by the norm of the components of the computation:

$$\varepsilon_R = \frac{\|x - y\|}{\max(\|x\|, \|y\|}.$$

Simplifying, no matter the processing that takes place, given the dynamic range $D$ required in the image, the weakest source that could be observed is $D$ times weaker than the brightest object. No other operation but averaging over samples can in any way reduce the absolute error. As it can be easily inferred, the relative errror is, so to speak, a ”sliding scale”: as the brightest objects are removed (cfr. CLEAN) the *absolute* error in the weakest sources will remain constant, but the *relative error* will increase, hence a reduction in the number of significant digits for them.

Errors are always to be computed normwise, unless specific methods, problems or analysis are invoked. The relative errors on individual elements are likewise meaningless: some values could be much smaller than others, carry the same absolute error yet their individual relative error could be very large: individual errors can be weighed but always by the problem norm (a collective quantity).

## 1.3 Significant Digits

*Significant digits* are the digits in a quantity that carry meaning: any other digits beyond them, represent just numerical “dirt” or “noise”, so to speak and nothing can be read into them. The mantissa only specifies the number of significant digits in a single number does not depend on the location of the decimal point. The number of significant digits could be exact, although limited: for example, the amount of money in my pocket right now (about ten pounds and pennies) has four significant digits (10.xx) and can be represented as a single precision number. Other times, a measurement may contain errors: a measuring tape allows measurement with one millimetre resolution: that does not mean, of course, that more accurate measurements cannot be taken, only that any digits representing fraction of a millimetre are meaningless.

In simple terms, significant digits represent the *information* that a number contains and at the same time represent the error (value of least significant digits relative to the value of the number)

As an example, the value of $\pi$ could be given as below.

| Value of $\pi$ | 3 | 3.1 | 3.14 | 3.142 |
|---|---|---|---|---|
| N. significant binary digits | 1 | 2 | 3 | 4 |
| Relative error | $5 \cdot 10^{-2}$ | $10^{-2}$ | $5 \cdot 10^{-4}$ | $5 \cdot 10^{-4}$ |

The number of significant digits provides an absolute, not a relative scale when numbers are combined. For example, when using 2 significant digits, $2.1 + 0.012 = 2.1$ with *two* significant digits, not $2.112$, which would have increased arbitrarily the number of significant digits to *four*: we have **no** information about the third and fourth digits of $2.1$ and we can make no assumptions at all about them. Also, by the same token: $2.1 - 0.098 = 2.0$.

Generally speaking, *normalizable* floating point numbers should be used (mantissa and exponent), apart, of course, from the quantized signals from the antennas. Normalizable numbers have the favourable characteristic that errors are, in general, normally distributed (rounding not truncating) and scaling to the appropriate decimal point are done automatically. Fixed point numbers (as well as, of course, integers) may cause problems by producing errors with non-Gaussian distributions, thus affecting results (and convergence) in ways which could then be difficult to correct.

Table 2 reports the number of digits (binary as well as decimal) for IEEE single and double precision floating point numbers. IEEE is specific in definition and behaviour of operations on such numbers and is virtually universally accepted. In my opinion, non-IEEE definitions and operations should be avoided because o their potential difficulties.

| **Precision** | **N. bits (binary digits)** | **N.Decimal Digits** | **Accuracy** |
|---|---|---|---|
| Single (4 Bytes) | $B_{SP} = 23$ | $6 \leq D_S P \leq 7$ | $\varepsilon = 2^{-23} \approx 1.2 \cdot 10^{-7}$ |
| Double (8 Bytes) | $B_{DP} = 52$ | $15 \leq D_D P \leq 16$ | $\varepsilon = 2^{-52} \approx 2 \cdot 10^{-16}$ |

**Table 2:** Floating point precision IEEE definitions

In practice, both in measurements and computation, errors are assumed to have normal distribution thus decreasing as the inverse square root of the number of samples. This point will be further expanded in a subsequent section.

# 2 Digits Gained and Lost

## 2.1 Samples and their distribution

We assume that antenna signal from sources (i.e. voltage) is a zero-mean Gaussian $\mathcal{N}(\mu, \sigma^2)$, $\mu = 0$, with the square of the standard deviation $\sigma^2$ equal to the source power; likewise error terms would be given by zero-mean Gaussian distribution $\mathcal{N}(0, \nu^2)$.

The source signal on one antenna or receiver is of course correlated to the signal on any other (adjustement for phase is here implied). Their product will be a positive definite distribution with mean $\sigma^2$, converging at a rate of the order of the inverse square root of the number of samples.

The noise distributions on the two antennas is of course uncorrelated and uncorrelated to the source signal. The product of two zero-mean uncorrelated Gaussian distributions asymptotically behaves like a normal distribution, with mean also converging to zero as the inverse square root of the number of samples.

## 2.2 Significant Digits Acquisition

Basically, an imaging pipeline, from input signals to image product(s) consists of a summation/averaging process, where a number of operations are carried out along the way.

We should also make clear at the onset that, in principle:

1. Only the summation of samples can add significant digits, there are no other mechanisms in the pipeline. And significant digits can only be added as the square root of the number of samples.

2. Numerical operations carried out on the measurements can only lead to the **loss** significant digits, never to their acquisition (that includes also macroscopic operations/procedures such as CLEAN).

3. significant digits lost cannot be re-acquired by any numerical operations but only by increasing the number of samples.

4. Any summation, e.g. over observation time, baselines, frequencies, etc. would reduce errors.

5. Most importantly: the maximum number of significant digits with respect to the brightest source in the observation. It is an *absolute not a relative* quantity. The dynamic range $D$ of course is just a measure of the significant digits acquired.

The last point is essential, albeit rather obvious: the removal of bright sources cannot add to the number of significant digits, it can only reduces them, however accurate the computation is. Weaker sources will simply be detected and studied with fewer significant digits.

For signals, noise, errors with Gaussian distribution, the number of significant bits in the mean and standard deviation increases with the logarithms (in base 2) of the square root of the number of samples $S$, namely $\log_2 \sqrt{S}$. This is obviously altogether equivalent to the reduction of the error as the inverse of $\sqrt{S}$.

Noise could be represented by significant digits loss in the input signal. Obviously, if noise is higher than signal, no significant digits are given by just one sample. So, noise can be considered, and will be so here, as ”negative” digits.

SKA antennas samples would be quantized to few bits: 8 bits for Low, 4-8 bits for Mid, depending on the band. The number of antenna signal bits would be defined at the onset by the need to avoid statistically significant clipping in the antenna signals and to make sure that enough bits are retained so as the signal of interest appears within the range of the number of bits, although this is not really needed. It is the SNR $\Xi$ (signal-to-noise ratio) which has the greatest impact on the convergence when the SNR is smaller than 1.

Hence, the average number of digits per sample $N_S$ could be given, in this very approximate model, by

$$N_S \sim \min(0, \log_2 \sqrt{\Xi})$$

using the symbols defined in Table 1.

The dynamic range required $D$ can be seen as the number of significant digits in the resulting image: for example, a $10^5$ dynamic range would require at least 5 significant decimal digits, equivalent to $\sim 17$ binary digits (bits). The removal of the stronger sources does not alter this.

Naturally, in an aperture array, the number of individual antennas need to be taken in account as well. The averaging of antennas within a station would increase the number of significant digits by $\sqrt{N_A}$.
During the imaging process, the visibilities of different baselines are also added together, thus providing further samples for the overall averaging and increasing the number of significant bits in the final image: an interferometer consisting of $N_R$ receivers has $\sim \frac{{N_R}^2}{2}$ baselines, attaining an approximate $\log_2(N_R/\sqrt{(2)})$ increase in the number of significant bits (binary digits). Averaging steps, include cross-correlation, any summation/averaging of time dumps before calibration etc., summation/averaging of visibilities, stacking independent images, summation/averaging of channels, etc.

We should also bear in mind that the convergence of the mean is not monotonic. Why should it be? It represents an asymptotic, average behaviour, and we cannot expect it to be followed exactly.

## 2.3 Significant Digits Loss

Any numerical computation carried out on a fixed set of data (i.e. excluding adding samples) incurs into loss of digits. No operation can reacquire lost digits.

Some operations, for example FFT, summations, averaging, various matrix and vector products cause, in general, minimal numerical error thus incurring in minimal loss of digits. Others, such as solvers, for example the solution of linear systems of equations, can leads to the loss of digits.

Suppose that we apply some Mathematical operation, for example, solve the system of simultaneous equations:

$$Ax = b$$

The relative (forward) error

$$\phi = \frac{\delta}{\zeta} = \frac{\|\bar{x} - x\|}{\|x\|}, \text{ where } \zeta = \|\bar{x}\|$$

where $\bar{x}$ is the exact solution of the problem caused by an operation can be estimated by:

$$\frac{\delta}{\zeta} \leq f(N)\ \kappa(\ldots)\ \varepsilon \tag{1}$$

where $\zeta$ is some normalization norm for the specific problem and the symbols are defined below in Table 3 below:

| Symbol | Description |
|---|---|
| $f(N)$ | Some (undefined) function of the problem size. In general, it varies very slowly with $N$. Pathological cases can be specifically built. |
| $\kappa(\ldots)$ | The condition number of the operation being carried out. This indicates the sensitivity of a solution obtained to small changes of the problem |
| $\zeta$ | The normalization factor from absolute to relative forward error. It is related to the norm of the problem. |
| $\|\ldots\|$ | Some matrix norm, such as 1-, 2- and infinity-norm; The Frobenius norm, although not a matrix norm is very often used because of its simplicity. |
| $\varepsilon$ | The machine accuracy. It is the smallest number such that $1 \neq 1+\varepsilon$. |

**Table 3:** Symbols used in the document

There is a great difference between *forward* and *backward* errors. The former estimates the distance between our solution and the exact theoretical solution; the latter just tells us if the solution satisfies the problem, i.e. residuals are small enough. In the case of the solution of simultaneous equations, the backward error would be given by:

$$\|Ax-b\| \leq \max(\|A\|\|x\|, \|b\|) f'(n)\varepsilon$$

where $f'(n)$ is some slowly varying function of $n$. Notice the absence of the condition number: that simply states that many methods can be backward stable but not necessarily in a forward sense (e.e. solution of linear equations).

The condition number is a very important concept in analysing numerical performance. It is intrinsic to the problem, and little that can be done to improve on it (i.e. to reduce it) without changing the computational workflow. Unfortunately, some algorithms can induce higher effective condition numbers, thus leading to greater digit losses. The oft-told butterfly effect describes, of course, an ill-conditioned problem. The worse (larger) the condition number, the more initial data (i.e. significant digits) are required to counteract loss of digits (i.e. information loss).

The normal equation method, very often used, has an effective condition number $\kappa(A)^2$ where $A$ is the matrix of the linear minimization system; other methods, such as SVD, QR, etc. have condition number $\kappa(A)$. So, for example, if $\kappa(A) \sim 10^4$, the normal equation method would stand to lose eight decimal digits; other methods may incur losses of four digits: single precision would be inadequate and higher precision be needed; the other methods could use single precision and return solutions accurate to three or four digits. However, for large residual problems, the effective condition number may depend on $\kappa(A)^2$, independent of the method.

Two further points need be raised at this point. The first is that digits are lost at the *least* significant end, of course. That means that if we have few significant digits, the effects can be limited to the numerically ”noisy” portion of the quantities of interest.

The second very important point is that the $\varepsilon$ factor applies to *fully converged* computations. When this is not the case, then we should write:

$$\frac{\delta}{\zeta} \leq f(N)\ \kappa(\ldots)\ \rho \tag{2}$$

where $\rho$ is the *residual*, i.e. the backwards error. That shows that ill-conditioned problems, i.e. those with high condition numbers, must be iterated until $log_{10}\rho \leq d_{10} + log 10\kappa$. For example, if $\kappa = 10^2$, and we have three significant digits, to avoid loss of digits, we would need to iterate until $\rho \leq 10^(-5)$ (actually, probably a bit less as digits are lost starting from the right). Of course, had we been using single precision and had five significant digits available, we would have needed to converge to $\rho \leq 10^{-7}$, an unattainable target: we would have either lost digits or have to recur to higher precision, such as double precision with its extended mantissa.

The table below summarizes the points above for some of the most common operations encountered in Radio Astronomy data reduction.

| **Operation** | **Normalization** $\zeta$ | $\kappa(\ldots)$ |
|---|---|---|
| Matrix vector product $Ax$ | $\|\|A\|\|\|\|x\|\|$ | $\sim 1$ |
| Sum of vectors $x+y$ | $\max(\|\|x\|\|, \|\|y\|\|)$ | $\sim 1$ |
| FFT of vector $x$ | $\|\|x\|\|$ | $\sim 1$ |
| Linear Solver, e.g. $Ax = b$ | $\|\|x\|\|$ | $\kappa(A) > 1$ ($\gg 1$ ?) |
| Linear Least Squares, $\min_x(\|\|Ax-b\|\|)$ | $\|\|A\|\|\|\|x\|\|\|$ | $\kappa(A) > 1$ ($\gg 1$ ?), see text |

**Table 4:** Values of the various symbols for some imaging pipeline operations

Matrix and vector operations could incur the loss of very few binary digits. Experiments with the FFT were carried out by the author and reported in an SDP Memo: this showed that binary digits loss is limited to 2 or 3 binary digits at most in cases likely to occur in SKA, including rather extreme cases.

A few more words are required by optimization (non-zero residual). In such case the backward error needs to be computed differently, and for *small* residual problems, the forward error $\phi$ is $O(\kappa)$; however for large residuals, it increases to $O(\kappa^2)$

## 2.4 Precision in phases and trigonometric functions

This really would require some comments. Trigonometric functions such as sines and cosines are periodical with $2\pi$ period. However, phase factors such as $2\pi kx$ where $x$ is some distance and $k$ the wave number occur throughout. Sines and cosines act upon $\mod_{2\pi}(2\pi k\, x)$ arguments, i.e. the modulo-$2\pi$ of their arguments. However, when $kx >> 1$ the modulo function causes significant loss of digits. For example, if $k = 10$ for a baseline of 100 Km ($10^5$ metres) 6 digits would be lost thus making the use of single precision impossible. However, it is possible to compute sines and cosines in single precision if $k, x$ are in double precision, and the double precision modulo function is employed. The modulo-$2\pi$ argument could then be transformed into single precision and sines and cosines computed in single precision.

This would not alter in any way the arguments presented in this document.

## 2.5 Putting it all together: the "economics" of significant digits

We have now all the ingredients to characterise the "economics" of the digits acquisition and loss for an observation, from the receivers' signals to the final image. This could provide guidance to the choice of precision required for the various steps.

Similarly to what is currently done in other fields, pipelines components could include monitoring facilities to allow changing dynamically (and automatically) the precision required. For example, solvers condition numbers could be tracked, varying the convergence tolerance etc. as required. For example, StefCal allows to estimate the condition number very cheaply as the iteration progresses, thus providing some estimate of the forward error. Other methods cannot easily do so.

In the table below, we report the contribution for each components of the "digits ledger book".

| Step | Value |
|---|---|
| Initial number of bits for each receiver's sample | $\min(0, \log_2 \sqrt{\Xi})$ |
| Gain: summation over all antennas in a receiver (station) | $\frac{1}{2} \log_2 N_A$ |
| Gain: summation over each calibration interval | $D_C = \frac{1}{2} \log_2 \frac{S T_C}{N_C}$ |
| Number of digits at this stage | $D_C = \min(0, \log_2 \sqrt{\Xi}) + \frac{1}{2} \log_2 \frac{S T_C N_A}{N_C}$ |
| Loss: solver (from condition number) | $D_S = \max[0, D_C + \log_2 \rho + log_2 \kappa]$ |
| Gain: overall observation summation | $\frac{1}{2} \log_2 \frac{T_O}{T_C}$ |
| Gain: summation/averaging over the baselines | $\log_2 \frac{N_R}{\sqrt{2}}$ |
| Loss: generic losses (FFT, gridding, etc.) | $L_G$ |

**Table 5:** Binary digits gain/losses for the computation stages

Finally, putting it all together, we can obtain $I_2$, the number of significant digits in the image:

$$I_2 = \log_2 D = \min(0, \log_2 \sqrt{\Xi}) + \frac{1}{2} \log_2 \frac{S T_C N_A}{N_C} - \max[0, D_C - (-\log_2 \rho - log_2 \kappa)] + \frac{1}{2} \log_2 \frac{T_O}{T_C} + \\ + \log_2 \frac{N_R}{\sqrt{2}} - L_G$$

where $D_C$, the number of significant digits, was described in Table 5.

This can be inverted to obtain the observation time in hours in terms of the required dynamic range:

$$H_0 = \frac{T_C 2^{[2(\log_2 D - D_C + D_S + L_G) - \log_2 \frac{N_R}{\sqrt{2}}]}}{3600}$$

## 2.6 An Example

Table 6 below show some examples for an instrument similar to the SKA LOW:

| Parameters | | | | |
|---|---|---|---|---|
| $N_A$ | N. antennas per station | 200 | 200 | 200 |
| $N_R$ | N. stations | 500 | 500 | 500 |
| $B_R$ | N. antenna bits | 8 | 8 | 8 |
| $\Xi$ | antennas SNR (power) | 1 | 0.1 | 0.01 |
| $N_C$ | N. channels | 65536 | 65536 | 65536 |
| $\kappa$ | Solver condition number | 10 | 10 | 10 |
| $L_G$ | N. bits lost in various ops (see text) | 4 | 4 | 4 |
| $T_C$ | Dump (Calibration) time intervals | 0.14 | 0.14 | 0.6 |
| $T_O(hours)$ | Observation time | 6 | 6 | 6 |
| $S(GHz)$ | Sampling rate | 1 | 1 | 1 |
| $\rho$ | Solver normalized residual | $10^{-5}$ | $10^{-5}$ | $10^{-5}$ |
| **Single Image: Number of significant digits (base 10)** | | | | |
| | Antenna input | 0 | $-1.1$ | $-3.3$ |
| | Before calibration/solver ($D_C$) | 3.1 | 2.6 | 2.1 |
| | After calibration/solver ($D_S$) | 3.0 | 2.6 | 2.1 |
| | After observation averaging (UV plane): at most | 4.8 | 5.0 | 4.5 |
| | Image (baselines averaging) | 6.7 | 6.4 | 5.9 |
| **Image Stacking: Number of significant digits (base 10)** | | | | |
| | Antenna input | 0 | $-1.1$ | $-3.3$ |
| | Before calibration/solver ($D_C$) | 3.1 | 2.6 | 2.1 |
| | After calibration/solver ($D_S$) | 3.0 | 2.6 | 2.1 |
| | Images (at each dump time): | 5.0 | 4.6 | 4.1 |
| | After stacking images | 6.7 | 6.4 | 5.9 |

**Table 6:** Some examples

Receivers calibration problems have in general low condition numbers, hence the low values of $\kappa$. It was also assumed that it is a *small residual* optimization problem, otherwise the loss of digits could be more significant.

The major impact on the number of digits is given by the SNR at the antenna. The number of antenna binary digits has some effect only when the SNR is very small.

The data and results above are purely indicative. They show that indeed this approach produces numbers, more or less, in the right ballpark. But this is not the point. the considerations here point reasonably clear to a strategy to switch between single and double precision at specific points of the computation.

Notice that in both cases, it is the final averaging (final steps in the table) that requires higher precision (at least in the case shown here). It would be possible to compute the previous steps using single precision.

Naturally, should the stacking be carried out at longer intervals, each image may well require accumulation in double precision, and double precision in the summation of images.

Basically, the switch to double precision would require single precision operands but accumulation in double precision.

# 3 Conclusions

From the previous discussion, it can be seen that only the averaging computational stages contribute to any increase in the number of significant digits. CLEAN major as well as minor cycles would not contribute to any increase in accuracy and would only result in a minor loss of digits.

Sources removal cannot add to the absolute accuracy, thus the relative accuracy would decrease, which is hardly surprising. This leads to an interesting and open question: should the computation reverts to single precision after the brightest sources have been removed?

This paper would like to suggest that a viable "prescription" could be, under likely SKA conditions:

| **Dynamic Range** | **Prescription** |
|---|---|
| $D < 10^5$ | Use single precision throughout |
| $D \geq 10^5$ | Use single precision until the last accumulation step, then switch to double precision using SP operands but accumulating in double precision, and DP for all subsequent steps. Should the computation revert to single precision after subtraction of the strongest sources, thus having reduced the number of significant digits? |

In all cases, as it has already been pointed out, great care must be taken in computing the antennas/receivers trigonometric factors to avoid loss of digits.

Although the effects of ill-conditioning when using solvers can be rather dramatic, simple steps can and should be taken

| **Prescription for ill-conditioned calibration** |
|---|
| Always carry out the solution to preserve the number of significant digits by correcting for the condition number. This simply requires setting the convergence tolerance appropriately. hline Monitor the condition number (it can be done trivially with some algorithms, e.g. StefCal) and correct the convergence requirements accordingly on-the-fly. hline Avoid normal equations and any other methods that induce higher condition numbers unless the condition number is demonstrably very low. |

# List of Tables