Detection and correction of silent errors in the conjugate gradient algorithm

Gérard MEURANT

CIRM, November 2021

Introduction

Today's supercomputers have more and more processing elements. Therefore, it is more likely than before that there can be some hardware failures during the execution of a user's parallel code

A silent error may cause an application floating point value to be changed to a different floating point value that is valid, but incorrect

This may have a strong and negative effect on the output of the code. The main problem is that these errors may emit no indication that something has gone wrong

The most basic technique for fault detection is to do redundant computations

These very general techniques are called double modular redundancy (DMR) and triple modular redundancy (TMR)
However, doing redundant computations is very costly and may be impracticable

A more interesting class of methods exploits the knowledge we may have about the numerical method in use
Early in the 1980s an algorithm was described capable of detecting and correcting a single silent error in a matrix-matrix multiplication using row and column check-sums

Other techniques for detecting silent errors in CG were

- monitoring the residual norms, from one iteration to the next or in the average, checking if the current residual norm is larger than the mean of the last m residual norms
- check of local orthogonality. For instance, check of $(p_{k-1}, Ap_k)/[\|p_{k-1}\| \, \|Ap_k\|]$ which must be zero in exact arithmetic
- comparison of the true and computed residual vectors

Monitoring the residual norms may not be easy when there are large oscillations of those norms and checking local orthogonality properties is not that obvious either

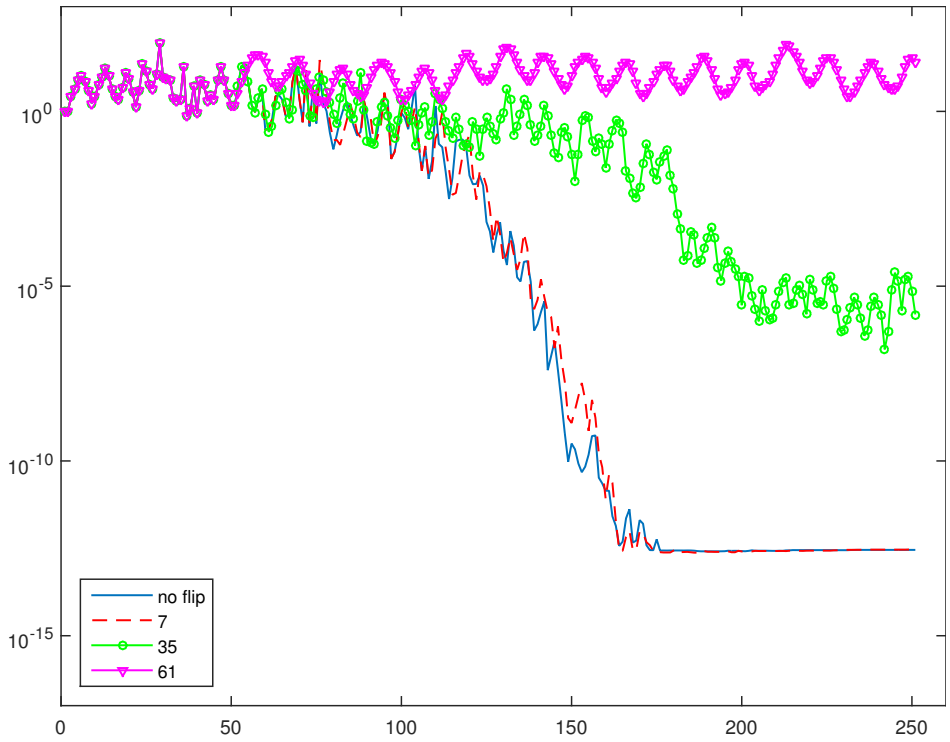
The effect of silent errors on CG convergence

In this work we simulate silent errors by bit flips in the output of some CG steps

Since we use IEEE double precision floating point arithmetic, the mantissa (or significand) has 52 bits that we number from 1 to 52, 1 being the rightmost bit, the exponent has 11 bits that is, bits 53 to 63 and the sign bit is bit 64

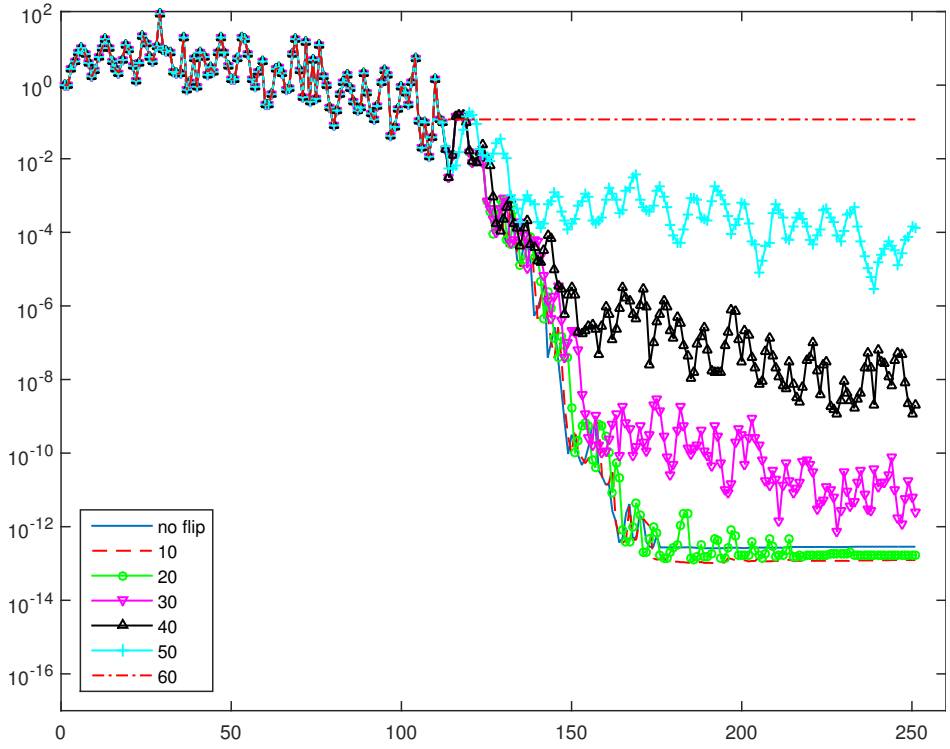
Intuitively we may think that flipping low order bits has less importance than, say, flipping a bit in the exponent field
It is also likely that flipping a bit when CG has almost converged is less dramatic than an early bit flip

Let us check this when solving a linear system with the matrix *bcsstk01*. This matrix is of order 48 with a condition number of $8.8234\ 10^5$. However, almost 220 CG iterations are needed to reach the maximum attainable accuracy for the true residual norm $\|b - Ax_k\|$. We use a zero initial vector

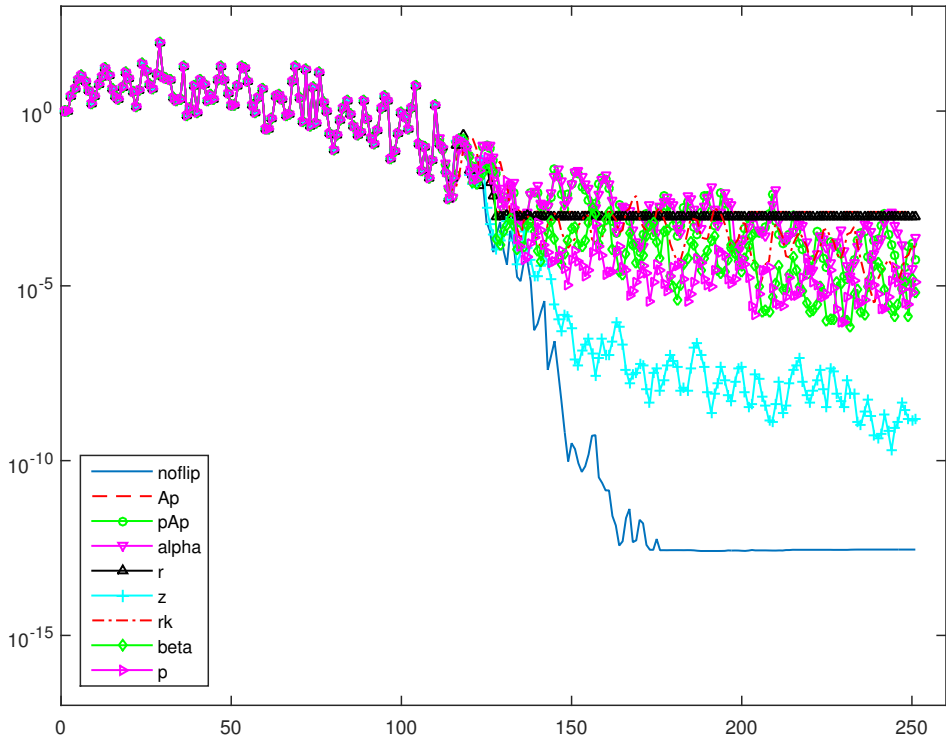


bcsstk01, relative true residual norms when flipping bits 7, 35 or 61 in (r_k, r_k) at iteration 50

Flipping the low order bit 7 does not do too much harm to CG convergence. Flipping bit 35 delays convergence and probably has a negative effect on the maximum attainable accuracy. CG does not converge any longer when flipping bit 61 which is in the exponent field



We successively flip bits 10, 20, 30, 40, 50 and 60 at iteration 110
Flipping bits 10 and 20 once do not change CG convergence.
Perturbation of bits larger than or equal to 30 delays or spoils convergence. For bits 30, 40 and 50 there is finally convergence but with a larger and larger delay. With the flip of bit 60 there is no convergence in 1000 iterations



Residual norms when we flip bit 60 in different CG steps at iteration 110

For vectors we flip the bit in only one component chosen randomly

We observe that in most cases there is no convergence

Detection of silent errors in CG

Let us consider a formula that was derived by Y. Saad in 1985 and used by G.M. in 1987 for the sake of introducing more parallelism in CG with a predictor-corrector technique

In CG the residual vectors satisfy the relation

$$r_k - r_{k-1} = -\alpha_{k-1} A p_{k-1}$$

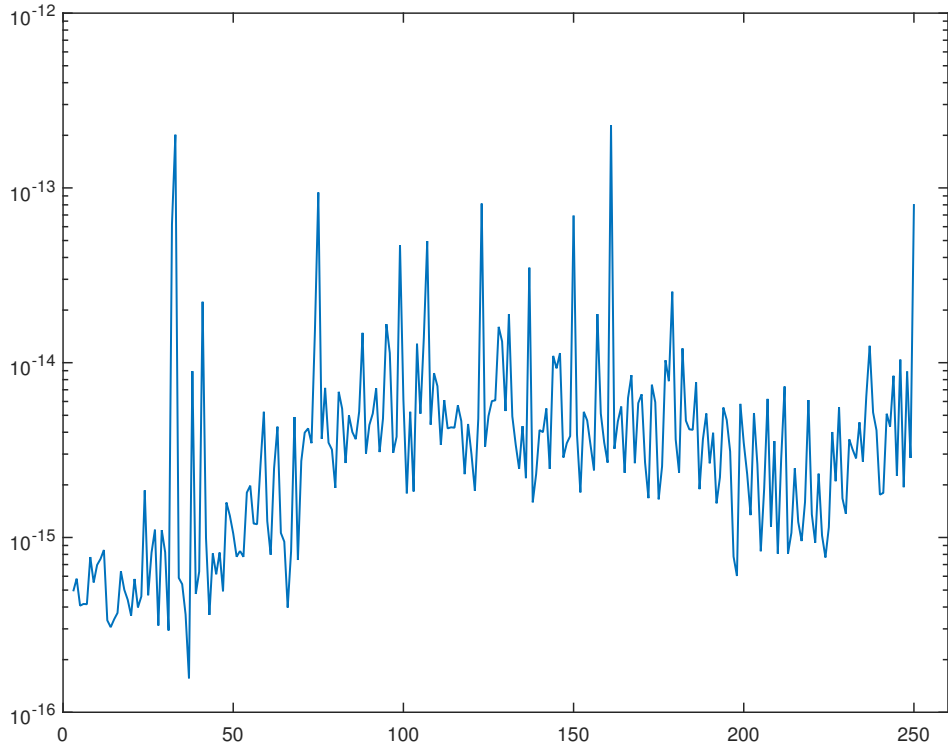
Using local orthogonality, we obtain

$$\alpha_{k-1}^2 (Ap_{k-1}, Ap_{k-1}) = (r_{k-1}, r_{k-1}) + (r_k, r_k) = \|r_{k-1}\|^2 + \|r_k\|^2 \quad (1)$$

We claim that if there are one or several silent errors in iteration k, this relation won't be satisfied

Let us define

$$d_k = \frac{|\alpha_{k-1}\|Ap_{k-1}\| - (\|r_{k-1}\|^2 + \|r_k\|^2)^{\frac{1}{2}}|}{(\|r_{k-1}\|^2 + \|r_k\|^2)^{\frac{1}{2}}}$$



Finite precision arithmetic

How is relation (1) satisfied in finite precision arithmetic?

$$\begin{aligned} r_k &= r_{k-1} - \alpha_{k-1} A p_{k-1} + \delta^r_{k-1} \\ p_k &= r_k + \beta_k p_{k-1} + \delta^p_{k-1} \end{aligned}$$

with

$$\begin{aligned} \|\delta^r_{k-1}\| &\leq u\|r_{k-1}\|\,(1 + C^r_{k-1}) + O(u^2) \\ \|\delta^p_{k-1}\| &\leq u\|r_k\|\,(1 + C^p_{k-1}) + O(u^2). \end{aligned}$$

The positive coefficients C^r_{k-1} and C^p_{k-1} are bounded because the ratios $\|r_{k-1}\|/\|p_{k-1}\|$, $\|p_{k-1}\|/\|r_{k-1}\|$ and $\|r_k\|/\|r_{k-1}\|$ are bounded

For relation (1) we obtain

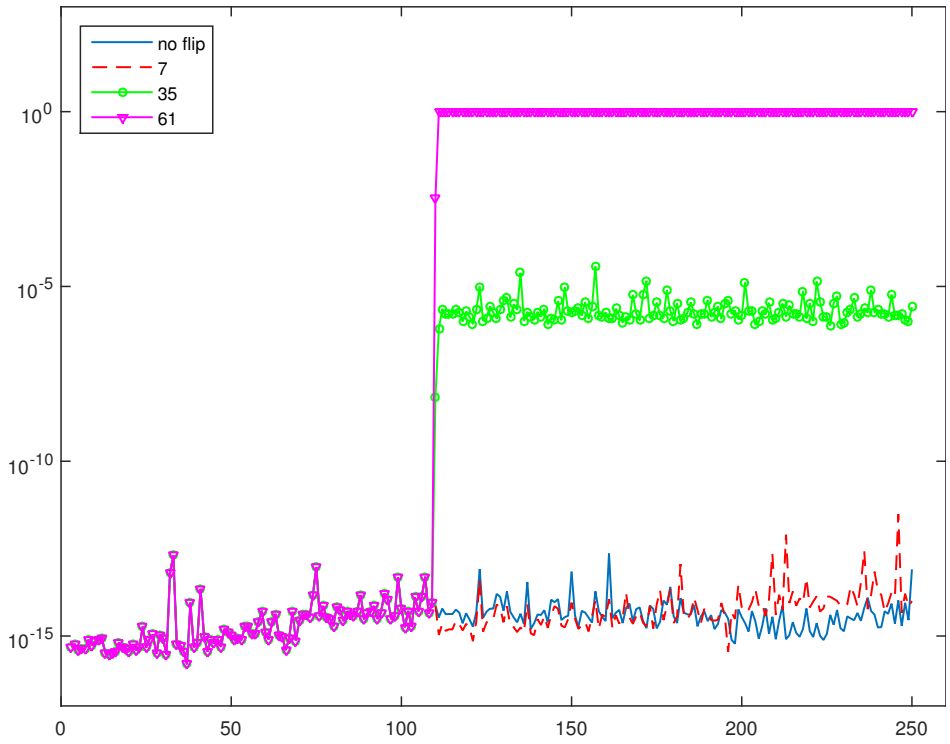
$$\begin{aligned}\alpha_{k-1}^2\|Ap_{k-1}\|^2 &= \|r_{k-1}\|^2+\|r_k\|^2+\|\delta_{k-1}^r\|^2\\ &+ 2\left[(r_{k-1}-r_k,\delta_{k-1}^r)-(r_{k-1},r_k)\right]\end{aligned}$$

It yields

$$\begin{aligned}\alpha_{k-1}\|Ap_{k-1}\| \leq (\|r_{k-1}\|^2 &+ \|r_k\|^2)^{\frac{1}{2}}+(\,\|\delta_{k-1}^r\|^2\\ &+ 2[(r_{k-1}-r_k,\delta_{k-1}^r)-(r_{k-1},r_k)]\,)^{\frac{1}{2}}\end{aligned}$$

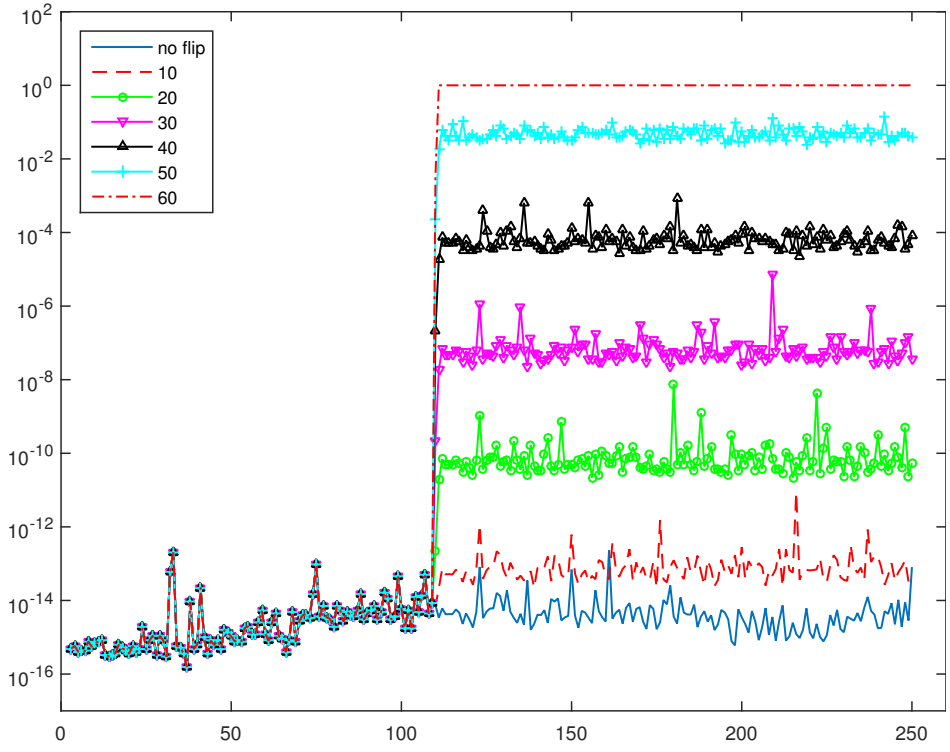
The term that can be a problem is

$$\frac{2|(r_{k-1},r_k)|}{\|r_{k-1}\|^2+\|r_k\|^2}$$



Relative difference d_k when flipping bits 7, 35 or 61 in (r_k, r_k) at iteration 110

The largest value of d_k is obtained when we flip bit 61 which is in the exponent field



Relative difference d_k when flipping bits in (r_k, r_k) at iteration 110

The larger is the bit number, the larger is the value of d_k

Detection of silent errors

Let ε_d be a given threshold

We say that there was a fault in the computation if $d_k > \varepsilon_d$

Note that there can also be a fault when computing d_k. However, the computation of d_k, which only involves a few scalars, can be done redundantly to be sure that the result is correct

With preconditioning and defining z_k by $Mz_k = r_k$, relation (1) becomes

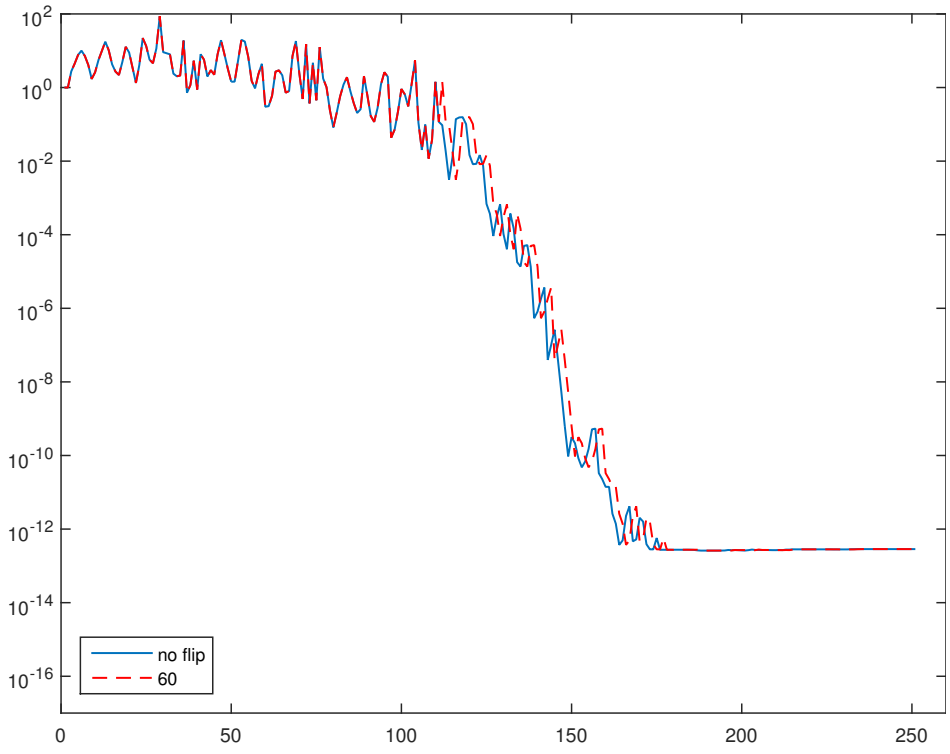
$$\alpha_{k-1}^2 (Ap_{k-1}, M^{-1}Ap_{k-1}) = (r_k, z_k) + (r_{k-1}, z_{k-1})$$

Correction of silent errors

Having a way to test if there was a silent fault at iteration k, we can implement a recovery procedure

Let us assume that we have at hand the values of the CG vectors and scalars at the beginning of iterations k and $k-1$

If we detect a fault, we go back to the beginning of iteration $k-1$
This is necessary because the fault could have happened in the computation of p_{k-1} at iteration $k-1$



Relative true residual norms when flipping bit 60 in (r_k, r_k) at iteration 110 with recovery

There is a delay when using the recovery procedure because we did not update the iteration number after the recovery

Predictor-corrector technique

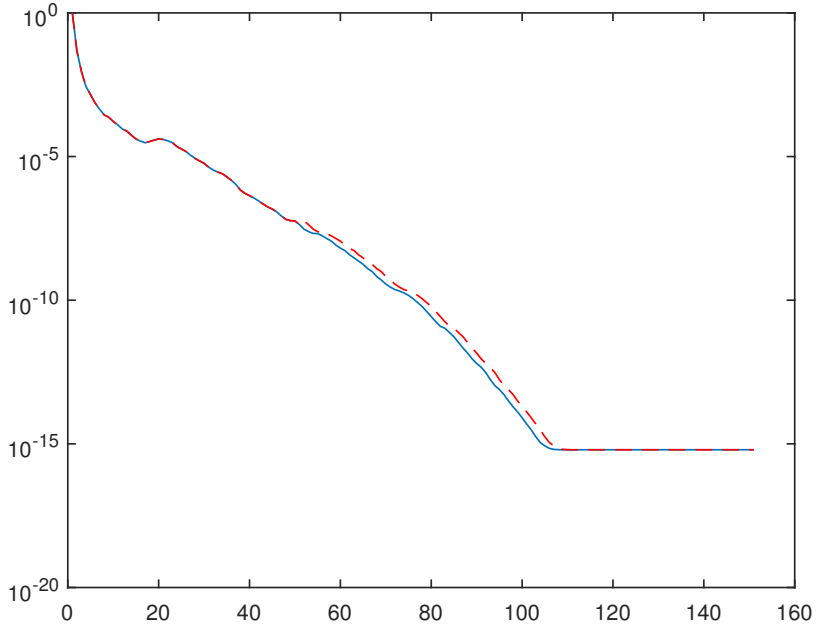
Relation (1) is used to predict the value of $r_k^T z_k$ which is used to compute

$$\beta_k = \frac{r_k^T z_k}{r_{k-1}^T z_{k-1}}$$

and p_k

The value of $r_k^T z_k$ is recomputed at the end of the iteration to prevent instability

This avoids one of the two solves with M in the detection process



kuu, incomplete Cholesky preconditioner, relative true residual norms when flipping bit 60 in (r_k, z_k) at iteration 50 with recovery

kuu is a matrix of order 7102 with a condition number $1.58\ 10^4$

Conclusions

We have proposed a new way to detect and correct silent errors in the conjugate gradient algorithm

The detection criterion is simple and cheap to implement

Numerical experiments show that the new criterion is robust and reliable

One of the advantages of our method, besides its simplicity, is that the check for errors can be done at every iteration, minimizing the extra computations that have to be done for the recovery

References

- Y. Saad, *Practical use of polynomial preconditionings for the conjugate gradient method*, SIAM J. Sci. Stat. Comput., v 6 n 4 (1985), pp. 865-881

- G. Meurant, *Multitasking the conjugate gradient method on the CRAY X-MP/48*, Parallel Computing, v 5, (1987), pp. 267-280

- G. Meurant, *The Lanczos and Conjugate Gradient algorithms, from theory to finite precision computations*, SIAM (2006)

- T. Chen and E. Carson, *Predict-and-recompute conjugate gradient variants*, SIAM J. Sci. Comput., v 42 n 5 (2020), pp. A3084-A3108

- E. Agullo, S. Cools, E. Fatih-Yetkin, L. Giraud, N. Schenkels and W. Vanroose, *On soft errors in the Conjugate Gradient method: Sensitivity and robust numerical detection*, SIAM J. Sci. Comput., v 42, n 6 (2020), pp. C-335-C358

Happy birthday Claude

CIRM 2007