As supercomputers grow ever larger in scale, the Mean Time to Failure becomes shorter and shorter, making the complete and successful execution of complex applications more and more difficult. **FT–LA delivers a new approach, utilizing Algorithm-Based Fault Tolerance (ABFT), to help factorization algorithms survive fail–stop failures.** The FT–LA software package extends ScALAPACK with ABFT routines, and in sharp contrast with legacy checkpoint-based approaches, ABFT does not incur I/O overhead, and promises a much more scalable protection scheme.

## ABFT: The Idea

### Protection

Matrix protected by block row checksum

The **algorithm** updates both the trailing matrix AND the checksums

\[ C_i' = \sum_j A_{ij} \]

### Recovery

**Missing blocks** reconstructed by inverting the checksum operation

\[ A_{ij} = C_i' - \sum_{k \neq j} A_{ik} \]

## Performance on Kraken

- Cost of ABFT comes only from extra flops (to update checksums) and extra storage
- Cost decreases with machine scale (divided by Q when using P\times Q processes)

## Functionality Coverage

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Systems of Equations</td>
<td>Cholesky, LU</td>
</tr>
<tr>
<td>Least Squares</td>
<td>QR (with protection of the upper and lower factors)</td>
</tr>
</tbody>
</table>

## Features

- Covering four precisions: double complex, single complex, double real, single real (ZCDS)
- Deploys on MPI FT draft (ULFM), or with "Checkpoint-on-failure"
- Allows toleration of permanent crashes

## Work in Progress

- Hessenberg Reduction, Soft (silent) Errors

---

FIND OUT MORE AT [http://icl.cs.utk.edu/ft-la](http://icl.cs.utk.edu/ft-la)