**XHYPRE: A HIGH-PRECISION NUMERICAL SOFTWARE PACKAGE FOR SOLVING LARGE-SCALE SPARSE LINEAR EQUATIONS**

**Chuanying Li**, **Stef Graillat**, **Hao Jiang**, **Zhe Quan**

1. Hunan University, Changsha, China
2. Sorbonne Université, CNRS, LIPI, Paris, France
3. National University of Defense Technology, Changsha, China

---

**Introduction**

When there are floating-point numbers with a limited number of digits representing real numbers, rounding errors will occur. Rounding error is unavoidable in floating-point operations, and it means the numerical difference between the exact value and the approximate value obtained by floating-point computation. We study how to control the cumulative effect of rounding errors because of the inaccurate results of numerical calculations on high-performance computing platforms.

In this paper, we adopt the error-free transformation technology to design and implement a high-precision numerical software package XHYPRE with the high-precision algorithms above for large-scale sparse linear equations. It is open-sourced at https://github.com/ compilerOpt/-XHYPRE-2.0.0.

**Methods**

- The high-precision dot product algorithm in XHYPRE uses the Dot2 algorithm [3].
- We improve the TwoProdFMA algorithm in the Dot2 algorithm to the TwoProd algorithm.
- The following will briefly explain the GMRES algorithm as an example. Algorithm 3 is a high-precision GMRES.
- The implementation process of PCG and BiCGSTAB is similar to the GMRES of high-precision, so we do not introduce them in detail.
- The input and output of the algorithm remain unchanged.

**Algorithm 2**

```
function Algorithm 2
Input: Matrix A, vector b
Output: Approximate solution of linear system Ax = b
1. Let r_0 = b - A*x
2. for i = 1 to max_itr do
   3.   r_i = r_0 + A*x
   4.   r_i = r_i + A*x
   5.   for j = 1 to max_itr do
      6.     x_i = x_i + A*x
      7.     x_i = x_i + A*x
   8.   end for
   9.   return x
```

**Algorithm 3**

```
function Algorithm 3
Input: Matrix A, vector b
Output: Approximate solution of linear system Ax = b
1. Compute r_0 = b - A*x
2. for i = 1 to max_itr do
   3.   Compute r_i = r_0 - A*x
   4.   Compute r_i = r_i - A*x
   5.   Compute r_i = r_i - A*x
   6.   return r_i
```

**Experiment**

Extensive experiments were conducted on the AMD platform, and the results illustrate that XHYPRE is effective.

**References**