BiCGSTAB

In numerical linear algebra, the biconjugate gradient stabilized method, often abbreviated as BiCGSTAB, is an iterative method developed by H. A. van der Vorst for the numerical solution of nonsymmetric linear systems. It is a variant of the biconjugate gradient method (BiCG) and has faster and smoother convergence than the original BiCG as well as other variants such as the conjugate gradient squared method (CGS).

Unpreconditioned BiCGSTAB

To solve a linear system Ax = b, BiCGSTAB starts with an initial guess x0 and proceeds as follows:

1. r0 = b − Ax0
2. Choose an arbitrary vector r0̂ such that (r0̂, r0) ≠ 0, e.g., r0̂ = r0
3. ρ0 = α = ω0 = 1
4. v0 = p0 = 0
5. For i = 1, 2, 3, …
    a. ρi = (r0̂, ri−1)

```
double roc=0;
        for (int i=1; i<=n; ++i) {
                double v1=0;
                if (r.getLine(i)!=NULL) v1=r.getLine(i)->val;

            roc+=v1*r1.getLine(i)->val;
          }
```

    b. β = (ρi/ρi−1)(α/ωi−1)

```
double beta=(roc/ro)*(alpha/w);
```

    c. pi = ri−1 + β(pi−1 − ωi−1vi−1)

```
p=r+(p-v*w)*beta;
```

    d. vi = Api

```
v=A*p;
```

    e. α = ρi/(r0̂, vi)

```
alpha=roc;
double sub=0;
for (int i=1; i<=n; ++i) {
          double v1=0;
          if (v.getLine(i)!=NULL) v1=v.getLine(i)->val;
            sub+=v1*r1.getLine(i)->val;
}
alpha/=sub;
```

    f. h = xi−1 + αpi

```
h=x+p*alpha;
```

    g. If h is accurate enough, then set xi = h and quit

```
double eBic=(A*h-b).norm();
if (eBic<=eSor) {
        x=h;
```

```
            break;
    }
```

h. s = ri−1 − αvi

```
    s=r-v*alpha;
```

i. t = As

```
    t=A*s;
```

j. ωi = (t, s)/(t, t)

```
    double sus=0, jos=0;
    for (int i=1; i<=n; ++i) {
            double v1=0;
            if (t.getLine(i)!=NULL)
                    v1=t.getLine(i)->val;
            double v2=0;
            if (s.getLine(i)!=NULL)
                    v2=s.getLine(i)->val;
            sus+=v1*v2;
             jos+=v1*v1;
    }
    w=sus/jos;
```

k. xi = h + ωis

```
    x=h+s*w;
```

l. If xi is accurate enough, then quit

```
    double eBic=(A*x-b).norm();
    if (eBic<=eSor) {
             ++k;
            Break;
     }
```

m. ri = s − ωit

```
    r=s-t*w;
    ro=roc;
```