An incomplete LU (ILU) factorization is generally computed by a Gaussian elimination process where certain nonzeros are neglected in some way. Here, we present a completely different method for computing ILU factorizations. The new method is iterative in nature and is highly parallel. Given a matrix $A$, the new method computes the incomplete factors $L$ and $U$ by using the property

$$(LU)_{ij} = a_{ij}, \quad (i, j) \in S$$

where $(LU)_{ij}$ denotes the $(i, j)$ entry of the product $LU$, $a_{ij}$ denotes the $(i, j)$ entry of $A$, and $S$ denotes a set of matrix locations where nonzeros are allowed. In other words, the factorization is exact on the sparsity pattern $S$. The new method interprets an ILU factorization as, instead of a Gaussian elimination process, a problem of computing the unknowns $l_{ij}$ and $u_{ij}$, which are the entries of $L$ and $U$, using property (1) as constraint equations. Such an approach may seem impractical because these equations are nonlinear and there are more equations than the number of rows in $A$. However, there are several potential advantages to computing an ILU factorization this way: 1) the equations can be solved using fine-grained parallelism, 2) the equations do not need to be solved very accurately to produce a good ILU preconditioner, 3) we often have a good initial guess for the solution, and 4) there is structure in the equations that make them easier to solve than one might expect.

The constraint equations may be solved via a fixed-point iteration, or a parallel asynchronous iteration in the parallel setting. We will present theoretical and experimental results on the convergence of these methods for computing ILU factorizations. The fact that the method is iterative means that the exact ILU factorization does not need to be computed, and we show experimentally that roughly converged factorizations can be very effective preconditioners. Furthermore, the number of iterations required is typically very small (less than 5). It is also easy to show theoretically that these iterations converge asymptotically.

Parallelizing the sparse triangular solves with the ILU factors is another challenge. Here, we advocate using Jacobi iterations to solve with the triangular factors, which is related to using Neumann series approximations which have also been advocated in the past, but have been found to be not robust. Such an approach also seems disadvantageous because the matrices are already triangular. However, just like in the iterative computation of the ILU factors, the motivation to use an iterative approach here is again parallelism and the lack of a need to solve these equations very accurately. Because the matrices are triangular, the Jacobi method is guaranteed to converge asymptotically. To improve robustness, a block Jacobi splitting may be used to improve the normality of the iteration matrix. Here, it is useful to find reorderings that produce diagonal blocks with large norms, but at the same time do not adversely affect ILU preconditioning.