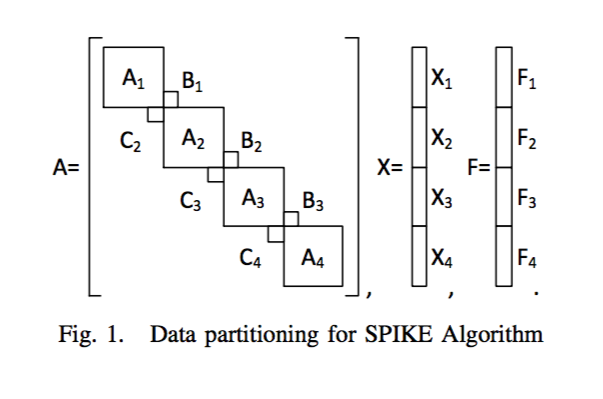
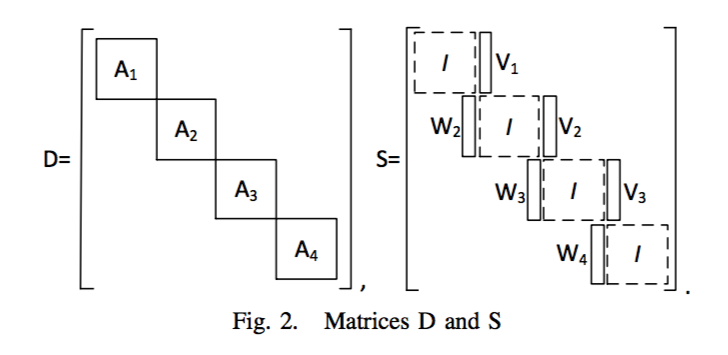
Computational fluid dynamics is a field of engineering that examines fluid motion using numerical methods and computational power. CFD has become an ubiquitous tool in industries ranging from aerospace to biomedical devices for its ability to accurately predict fluid phenomena in a variety of applications. However, as the complexity and precision of these applications increases the computational time required for accurate solutions exponentially grows. One possible solution to this challenge is to solve CFD problems in parallel using graphic processing units.

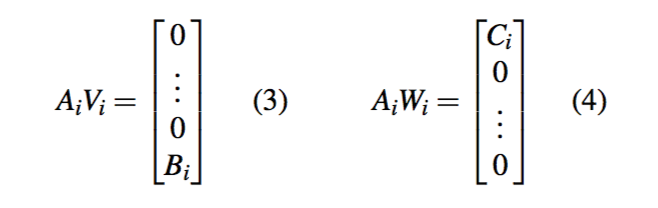
A typical CFD problem creates fields for pressure and velocity using the well-known Navier Stokes equations. These fields consist of a number of individual nodes that are governed by equations resulting from the Navier Stokes equations. The complete Navier-Stokes equations are solved using the SIMPLER method. The continuity equation is turned into an equation for the pressure correction and a pressure field is derived from the known velocity field. Each iteration, the velocities are corrected using velocity-correction formulas. The computations proceed to convergence via a series of continuity satisfying velocity fields. The algebraic equations are solved using a TDMA (Tri-Diagonal Matrix Algorithm) algorithm.

The algebraic equations solved using the TDMA are an ideal place to improve the computational time of a particular CFD problem[[1](#_ENREF_1)]. Since these equations are solved serially, a large percentage of the computational time is based on the clock-speed of the processor. However, these algebraic equations can be distributed over the massive number of processors on a graphics processing unit (GPU) and calculated in parallel resulting in a significant reduction of the overall computational time.

The tri-diagonal matrix can be solved in parallel using the SPIKE algorithm, a hybrid parallel solver for banded linear systems, developed by Polizzi et. al [[2](#_ENREF_2)]. The SPIKE algorithm starts with a preprocessing stage that partitions a tri-diagonal matrix of the form Ax = B, where A is the tri-diagonal matrix and x and B are vectors, to several square diagonal blocks. Furthermore, the diagonal components of the square blocks can be isolated by defining a matrix A = DS, where D is the isolated diagonals and S is a matrix of tri-diagonal form with diagonals of value equal to 1. These matricies can be seen below courtesy of Chang et al. [[3](#_ENREF_3)].







Next, vector Y can be created from SX and vector F can be created from DY. Since D is a block-diagonal matrix, the tiles of the system are independent and can be rearranged as AiYi = Fi. Since Ai and Fi are known, Yi can be solved for. Furthermore, Bi and Ci are known allowing Vi and Wi to be solved. Each of these equations must be solved serially, but clusters of the block diagonal matrix can be computer in parallel. Therefore, as the tri-diagonal matrix increases in size the computational time can be significantly reduced by distributing the reduced matrix blocks. Each of the small computational blocks will be sent to a separate processor on the GPU using an optimization program developed by GPU manufacturer NVidia. The implementation of the SPIKE algorithm has been common in multiple CFD applications in recent years with impressive results [[4](#_ENREF_4)].

1. Hang Liu; Jung-Hee Seo; Mittal, R.H., H.H., *GPU-accelerated scalable solver for banded linear systems*, in *2013 IEEE International Conference on Cluster Computing (CLUSTER)*. 2013, IEEE: Indianapolis, IN. p. 1-8.

2. Sameh, E.P.A., *A parallel hybrid banded system solver: the SPIKE algorithm.* Parallel Comput, 2006. **32**(2): p. 177-194.

3. Li-Wen Chang, J.A.S., Hee-Seok Kim, Wen-Mei W. Hwu, *A Scalable, Numerically Stable, High-performance Tridiagonal Solver using GPUs*, in *SC12*, IEEE, Editor. 2012, IEEE: Salt Lake City, Utah, USA.

4. Crespo, A.C., et al., *GPUs, a new tool of acceleration in CFD: efficiency and reliability on smoothed particle hydrodynamics methods.* PLoS One, 2011. **6**(6): p. e20685.