

# Molecular Electrostatic Potential Evaluation with the Fragment Molecular Orbital Method

Yuri Alexeev<sup>1</sup>, Dmitri G. Fedorov<sup>2</sup>

<sup>1</sup>Leadership Computing Facility, Argonne National Laboratory (USA); <sup>2</sup>National Institute of Advanced Industrial Science and Technology (Japan)

## Abstract

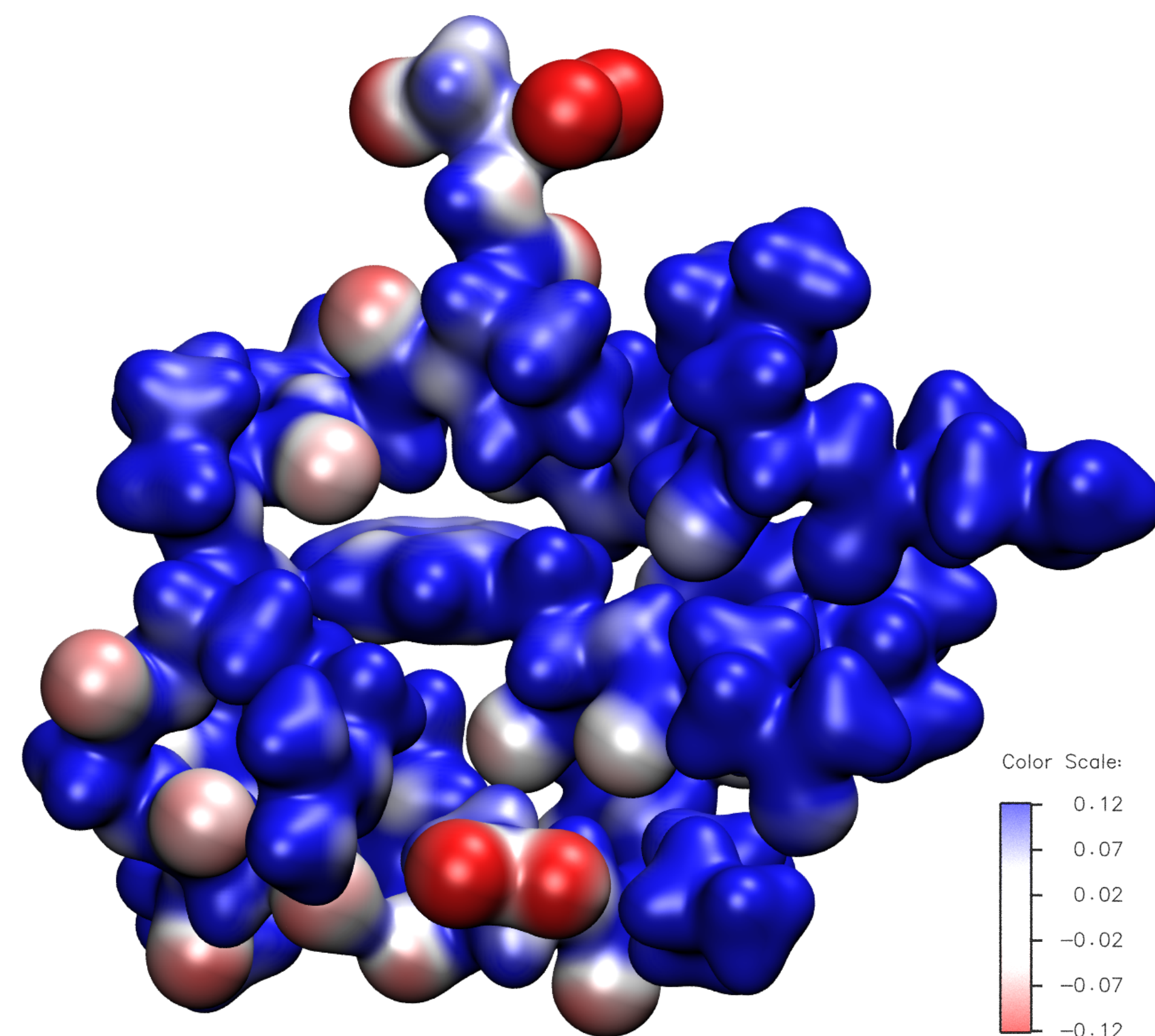
The molecular electrostatic potential (MEP) is a useful tool to analyze intermolecular electrostatic interactions and the properties of the chemical system. The most accurate way to compute MEP is to use quantum mechanics methods, but it is prohibitively computationally expensive for large chemical systems. Presently, the ability to compute MEP accurately for large systems is in high demand because of the recent advances in X-ray, cryo-electron microscopy, NMR, and mass-spectrometry techniques for elucidation of structure and conformation. The solution is to use linearly scaling QM methods, like fragment molecular orbital (FMO) method. The major problems are accurate computation of MEP, the storage of electron density and electrostatic potential in memory, and scalability of the code. To address these issues, we implemented different MEP algorithms and compared their performance. It was found that the new fragment cube method (FCM) produces accurate MEP at a fraction of cost.

## Supercomputer Challenge

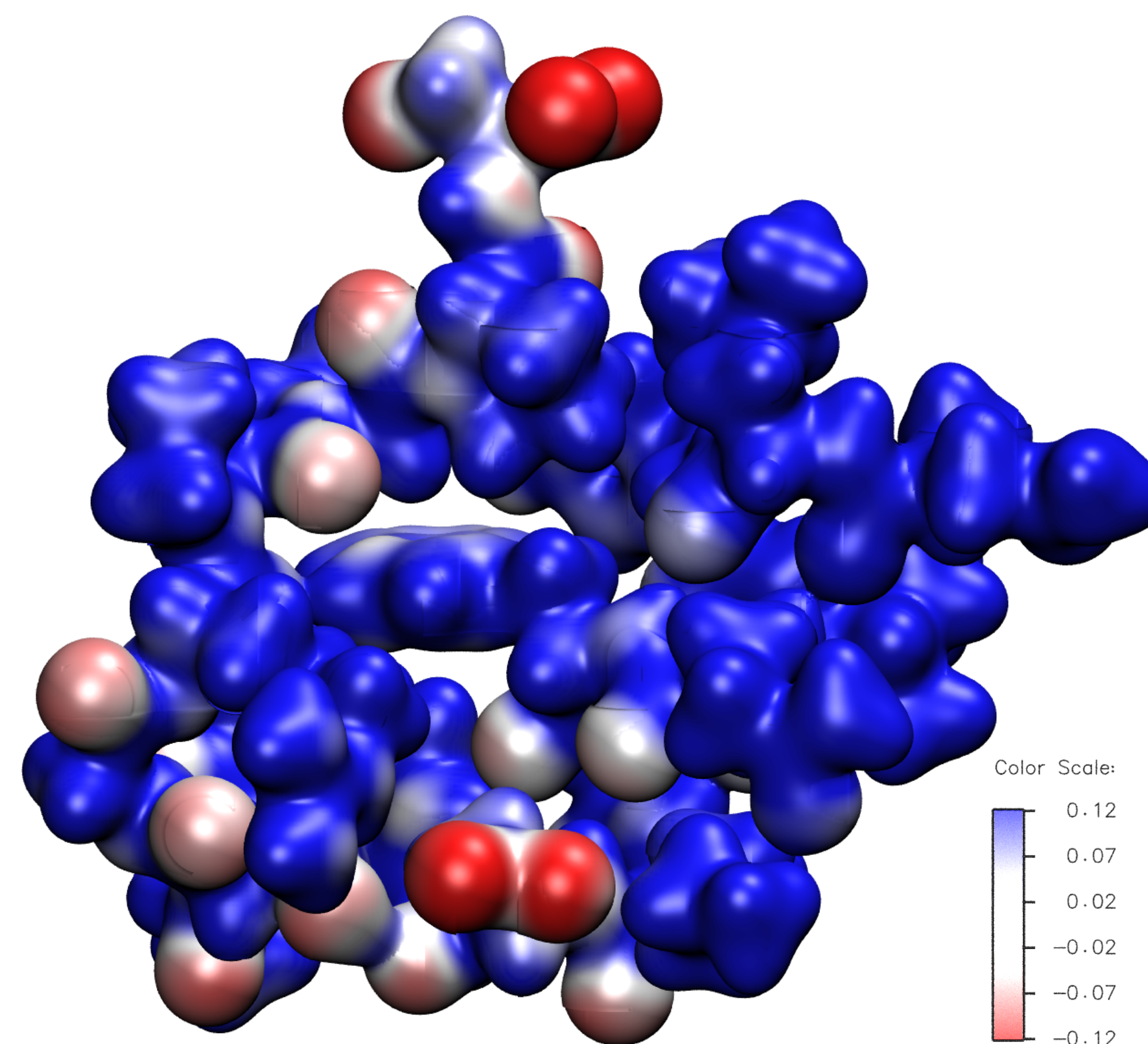
ALCF supercomputer Mira Blue Gene/Q architecture:

- 1,024 nodes per rack
- 16 cores/node
- 1.6 GHz processor
- 16 GB of memory/node
- Mira has a total of
- 48 racks (768,000 cores)
- 768 terabytes of RAM
- peak performance of 10 petaflops
- 384 IO nodes
- 240 GB/s, 35 PB storage
- ALCF PC cluster Cooley Intel Haswell architecture
- 126 compute nodes
- 12 CPU cores and one NVIDIA Tesla K80 dual-GPU card per node
- 2.4 GHz processor
- 384 GB of memory/node

A challenge for quantum chemistry codes such as GAMESS is to efficiently use the enormous number of cores which require well parallelized and scalable algorithms. For example, in this work a number of parallelization of algorithms were attempted for efficient parallelization of MEP code.



MEP figure computed with TCM. MEP is computed on electron density isosurface cutoff 0.02. MEP is computed for GRDPAD=1,1,100. Compared to TCM there are a few barely noticeable stitches on the fragment cube boundaries.



MEP figure computed with FCM. MEP is computed on electron density isosurface cutoff 0.02. MEP is computed for GRDPAD=1,1,100. Compared to TCM there are a few barely noticeable stitches on the fragment cube boundaries.

## Molecular Electrostatic Potential (MEP) method

There are two ways to compute MEP:

1. Solve Poisson-Boltzmann equation van der Waals molecular surface: The first step is to define van der Waals molecular surface which is defined as a union of spherical atom surfaces where each atom surface is defined by the van der Waals radius. Thus it is an approximation of the surface. The second step is to solve Poisson-Boltzmann equation

$$-\nabla \epsilon(x) \nabla \phi(x) + k^{-2} \sinh \phi(x) = f(x)$$

where  $\phi$  is the electrostatic potential,  $\epsilon$  is the dielectric constant of solute and solvent,  $k^2$  is the ionic strength of the solution and the accessibility of ions to the solute interior, and  $f$  describes the distribution of atomic partial charges of solute.

2. A more accurate way to compute MEP is with QM methods. Firstly, electron density is computed. Next, electron density isosurface is extracted from density with a typical cutoff value of 0.002, which captures about 99% of electron density. As a result, the surface which defines the molecule for MEP is more accurate than in PBE approach. The second step is computing electrostatic potential with a QM method by using this formula

$$\phi(\mathbf{r}) = \sum_A \frac{Z_A}{|\mathbf{r}-\mathbf{R}_A|} - \sum_{\mu\nu} D_{\mu\nu} \int \chi_{\mu}(\mathbf{r}') \frac{1}{|\mathbf{r}-\mathbf{r}'|} \chi_{\nu}(\mathbf{r}') d\mathbf{r}'$$

where  $Z_A$  is the nuclear charge of Atom A,  $\mu$  and  $\nu$  are atomic orbital basis functions.  $D_{\mu\nu}$  is the one-electron density matrix and it is defined in the following way:

$$D_{\mu\nu} = 2 \sum_i \phi_{\mu} \phi_{\nu}$$

## Fragment Molecular Orbital (FMO) method

In FMO, a molecular system is divided into fragments and the total properties, such as the energy or its gradient, are calculated from those of the fragments and (in FMO2) their pairs, computed in the embedding potential. FMO treats the electrostatics for the whole system while the exchange-repulsion and charge transfer are only accounted for at the local level of fragment pairs.

The total energy can be written as:

$$E = \sum_I E_I + \sum_{I < J} (E_{IJ} - E_I - E_J)$$

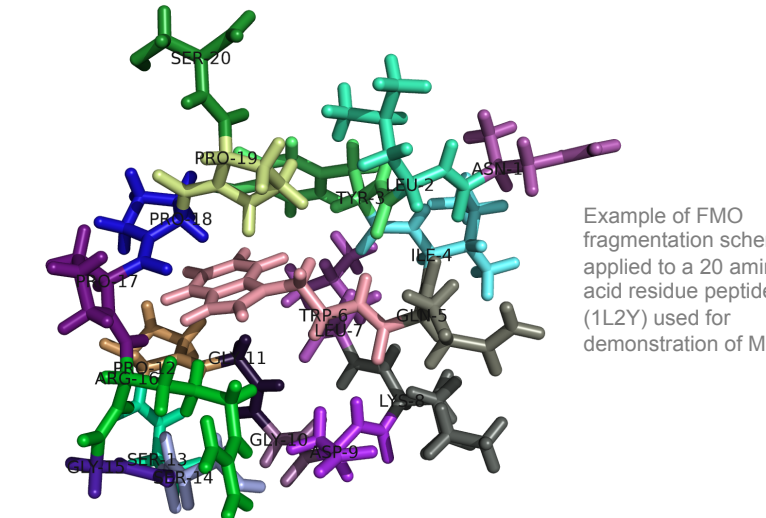
where the monomer (I), dimer (IJ) [FMO2] energies can be obtained using any standard QM method.

Current applications:

- Geometry optimization (~2,000 atoms)
- *Ab initio* level single point energies (~20,000 atoms)
- Pair interaction analysis (~20,000 atoms)
  - Drug design, ligand docking, chemical reactions in solution, etc.

Massively parallel FMO utilizes the Generalized Distributed Data Interface (GDDI)

- With the molecule divided into fragments, each fragment or pair is computed on a distinct subset, or group, of the available processors.
- Each fragment/pair is then run in parallel in each group.
- This provides two levels of parallelization, greatly increasing the parallel efficiency.



Example of FMO fragmentation scheme applied to a 20 amino-acid residue peptide (1L2Y) used for demonstration of MEP.

## FMO MEP Algorithms

There are two ways to compute MEP with FMO: total cube method (TCM) and fragment cube method (FCM). It is important to note that all calculations are done with FMO1 i.e., there are no fragment dimer calculations. FMO1 takes into account many-body polarization by using the embedding potential dependent on the electronic state of fragments, and it also takes into account intra- (but not inter-) fragment charge transfer.

## Total Cube Method (TCM)

### TCM Algorithm

```

Do
  For Ifrag=1,Nfrag
    Load balancing
    Compute electron density
    Broadcast densities/charges
  End
  While not converged monomer densities
  For Ifrag=1,Nfrag
    Load balancing
    For I=1,Nshell
      For J=1,I
        For k=1,ngrid_TCM(Ifrag)
          Compute ESP  $\phi_{Ifragk}$ 
           $\phi_{Ifragk} = \phi_{Ifrag} + \phi_{Ifragk}$ 
        End
      End
    End
  End
End
    
```

The whole system is put into a cube by leaving an extra space of GRDPAD(1) around the atoms.

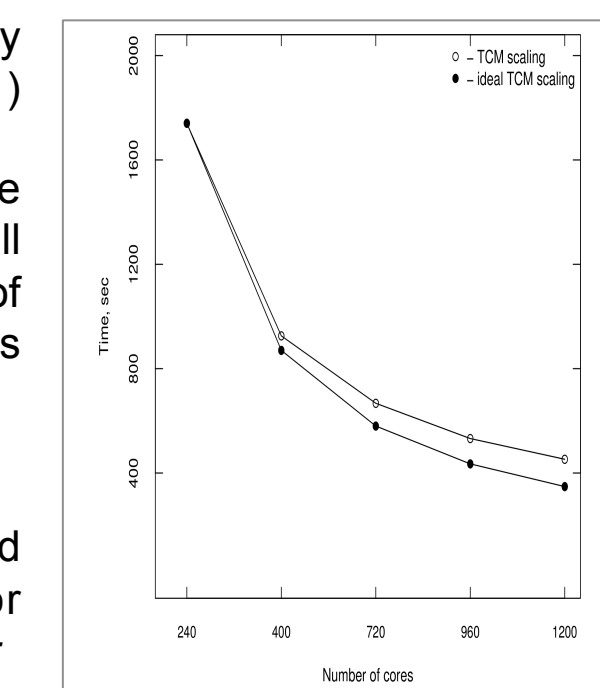
For each fragment  $I$ , a window in the total grid is determined, including all atoms in  $I$  and an extra space of GRDPAD(2). A subset of all grid points  $\mathbf{r}$  in the window is denoted as  $\mathbf{r}'$ :

$$\phi^I(\mathbf{r}) = \sum_A \frac{Z_A}{|\mathbf{r}-\mathbf{R}_A|} - \sum_{\mu\nu} D_{\mu\nu} \int \chi_{\mu}(\mathbf{r}') \frac{1}{|\mathbf{r}-\mathbf{r}'|} \chi_{\nu}(\mathbf{r}') d\mathbf{r}'$$

for  $\mathbf{r}$  in  $\mathbf{r}'$

Then, the total data are accumulated as by summing contributions for fragments  $I$ , whose window included  $\mathbf{r}'$ :

$$\phi(\mathbf{r}) = \sum_{I=1}^N \phi^I(\mathbf{r})$$



## Fragment Cube Method (FCM)

### FCM Algorithm

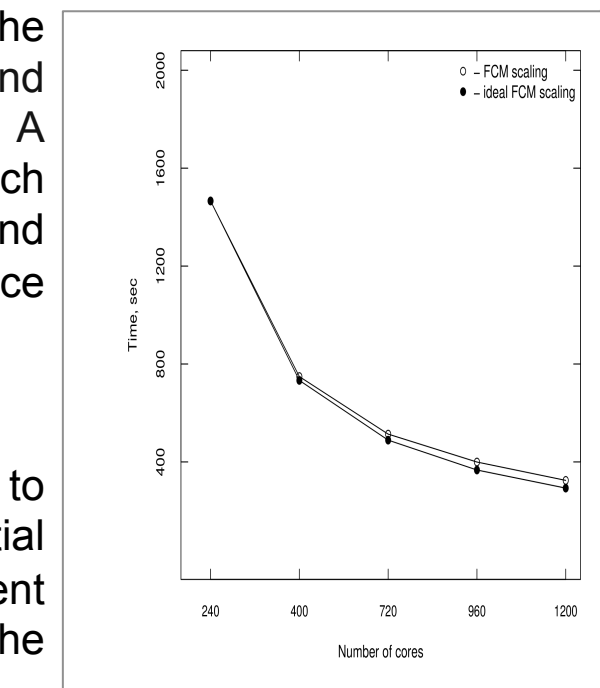
```

Do
  For Ifrag=1,Nfrag
    Load balancing
    Compute electron density
    Broadcast densities/charges
  End
  While not converged monomer densities
  For Ifrag=1,Nfrag
    Load balancing
    For I=1,Nshell
      For J=1,I
        For k=1,ngrid_FCM(Ifrag)
          Compute ESP  $\phi_{Ifragk}$ 
           $\phi_{Ifragk} = \phi_{Ifrag} + \phi_{Ifragk}$ 
        End
      End
    End
  End
End
    
```

The algorithm for computing MEP is the same. The difference between FCM and TCM is in the storage of cubes. A separate cube file is created for each fragment by constructing a cube around the fragment leaving an extra space GRDPAD(1). MEP is computed as

$$\phi(\mathbf{r}') = \sum_{I=1}^N \phi^I(\mathbf{r}')$$

meaning that the contribution due to fragments  $J$  is added to the potential computed for the cube around fragment  $I$ . GRDPAD(3) defines  $R_{\text{cube}}$ .  $R_{IJ}$  is the interfragment distance in FMO.



The principal difference between TCM and FMO is the way the potential and electron density arrays are computed, stored in memory, and later written to the cube files. In TCM, the total potential of the whole molecule needs to be computed and stored. There is an inefficiency in computing potential and density since there will be empty contributions (mostly coming from the corners of the cube and other empty spaces) and steep memory requirements to store the whole potential/density in memory, and writing large arrays to cube files. The alternative is FCM where potential and density are computed for each fragment individually. In FCM, it is possible to generate the total density and potential by merging/stitching fragment cube files. As a result, FCM compared to TCM runs faster, scales better, and requires a fraction of RAM memory and disk space while producing similar in quality density and MEP.

## Results

The calculations have been done on PC cluster Cooley for a small tryptophan cage protein (PDB code: 1L2Y) by using FMO1 with RHF and 6-31G\* basis set.

## Conclusions

In this work, we developed and parallelized two new algorithms to compute and store MEP of large chemical systems which can be computed with FMO. We found that FCM is a more scalable solution and it also provides the shortest time to solution compared to TCM. Both TCM and FCM are significantly faster than MEP calculations with traditional QM calculations, at the same time TCM/FCM are more accurate than MEP computed with APBS.



## Mira and Cooley used for DOE-run INCITE and ALCC projects

Innovative and Novel Computational Impact on Theory and Experiment (INCITE) and Argonne Leadership Computing Challenge (ALCC) programs aim to accelerate scientific discoveries and technological innovations by awarding, on a competitive basis, time on supercomputers to researchers with large-scale, computationally intensive projects that address "grand challenges" in science and engineering. 2015 INCITE Allocations at ALCF: 3.57 billion core-hours on Mira.

## GAMESS

(General Atomic and Molecular Electronic Structure System)

- *Ab initio* quantum chemistry package.
- Maintained by the research group of Prof. Mark Gordon at Iowa State University (<http://www.msg.ameslab.gov/gameSS>).
- Enables most major quantum mechanical methods (Hartree-Fock, Møller-Plesset perturbation theory, coupled cluster, multiconfiguration self consistent field, configuration interaction, density functional theory).
- Ported to most major computer architectures.
- Free and widely used on everything from laptops to supercomputers.
- About a million lines of code, with an associated parallelization library comprising 15,000 lines.
- Highly scalable, including many distributed data algorithms.