Development of a Discrete Mass Inflow Boundary Condition for MFIX

Jordan Musser
West Virginia University
Morgantown, WV 26506

Mary Ann Drumright-Clarke
West Virginia University
Morgantown, WV 26506

Janine Galvin
National Energy Technology Laboratory
Albany, OR 97321

ABSTRACT

MFIX (Multiphase Flow with Interphase eXchanges) is an open source software package developed by the National Energy Technology Laboratory (NETL) used for modeling the chemical reactions, heat transfer, and hydrodynamics of fluid-solid systems. Currently, the stable publically available release of MFIX does not include a discrete mass inflow boundary condition (DMIBC) for its discrete element method (DEM) package. Inflow boundary conditions are useful for simulating systems where particles are consumed through chemical reactions and an incoming feed is necessary to sustain the reaction. To implement the DMIBC an inlet staging area is designated outside the computational domain and particles are passed through the wall region associated with the inlet. Forces incurred on entering particles, generated from collisions with particles already in the system, are ignored whereas, particles already in the system respond to contact forces and react accordingly, moving away from the inlet. This approach prevents any unphysical overlap between new and existing particles. It also ensures that particles entering the system will enter the computational domain regardless of opposing forces. Once an incoming particle is fully within the domain, it reacts appropriately to any and all contact force. This approach for a DMIBC has been implemented and is available within the current development version of MFIX.

Keywords: Discrete Element Method (DEM), Discrete Mass Inlet Boundary Condition (DMIBC)

INTRODUCTION

The DEM is used in the modeling of granular assemblies and gas-solid flows like those found in the pharmaceutical, mineral processing, and energy industries. Unlike continuum models of solids, DEM tracks the position, velocity, and force statistics for every particle.

This work expands the DEM package initially implemented in MFIX by Boyalakuntla[1]. The original implementation was designed to operate over a static number of particles specified at the start of a new simulation. The current modifications allow for a more dynamic system by permitting particles to enter and exit the computational domain during a simulation through user-defined boundary conditions. This feature is desirable when considering models where particles are recycled or simulations where particles need to be replenished after being consumed through a chemical reaction and/or a phase change. This work is in concert with a larger effort to include heat and mass transfer between the Eulerian and Lagrangian models of MFIX.

Throughout this paper, DEM refers to the soft sphere model first proposed by Cundall and Strack[2]. In their model, springs, dash-pots, and sliders are used to account for the elastic, damping, and rotational forces generated during particle-particle collisions. As particles collide, the deformation of each particle's shape is captured through the overlap of contact points. The magnitude of the overlap is related to the scale of forces resulting from impact. As the collision evolves, the forces are used to recalculate the positions, trajectories, and rotational speeds of the particles. A detailed review of the DEM implementation within the stable release of MFIX is given by Garg, Galvin, Li, and Pannala[5].

Since the DEM uses the temporary overlap of particles to determine the kinetic response to a collision, an incoming particle cannot be initially placed overlapping another particle. This limitation becomes apparent if a DMIBC is defined along a domain region that is associated with high particle density, such as a packed bed. To avoid improper placement of particles while retaining the ability to define the DMIBC along any region of the domain, a staging area is designated within the ghost cells bordering the location of the DMIBC (Fig. 1).

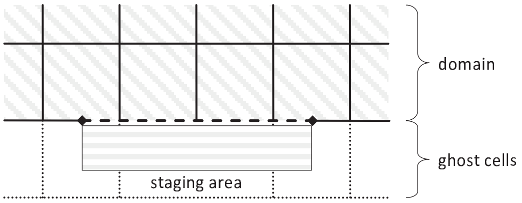

Figure 1: The staging area for incoming particles is located outside of the computational domain, in the ghost cells bordering the location identified with the boundary condition.

Particles are seeded into the staging area and then incrementally moved across the boundary into the domain of the system. The behavior of particles within the staging area is controlled so that given acceptable boundary condition parameters, a new particle can always be introduced without overlapping particles that have already been seeded.

As a particle passes from the staging area into the computational domain, contact forces acting on the incoming particle are temporarily dismissed but particles already within the system respond accordingly. Thus, entering particles will move particles obstructing their entry out of the way, despite any disparities in size or density. Once an incoming particle has fully entered the domain, it then responds to any forces acting on it in full accordance to the DEM. Here it is assumed that since there is a net flux of particles into the system and the solids fraction near the inlet region will be relatively low (i.e. not packed) for most situations [3,4], the overall effect of incoming particles neglecting contact forces is negligible.

This approach has been fully implemented in the development version of MFIX, with all code written in FORTRAN 90.

DISCRETE MASS INLET BOUNDARY CONDITION

The DMIBC has two operating components. The first element is called during the initialization of the MFIX software, wherein user provided data is checked for consistency, and operational characteristics are determined. The second element is executed during a full MFIX simulation, and controls the placement and assignment of physical properties to incoming particles.

Boundary Condition Initialization

To define the operational characteristics of a DMIBC, the physical properties and feed rates of the incoming particles are specified. The physical properties of a particle are identified through its mass phase, with each phase characterized by the diameter, d_m, and material density, ρ_m, of the particles belonging to that phase. Note, particles belonging to the mass phase indexed by m are identified by a subscript m and are referred to as 'phase-m' particles. The feed rate for each mass phase is given by a volumetric flow rate and bulk density, $\dot{\mathcal{V}}_m$ and ρ_{bm}, respectively. Here ρ_{bm} is the product of material density and the volume fraction of phase-m particles.

The flow properties for each mass phase are then discretized. This is done by converting the phase-m volumetric flow rate into an equivalent number of phase-m particles entering the system per unit time, $N_{m,t}$. The total number of incoming particles per unit time, N_t, is given by summing $N_{m,t}$ over all mass phases. In Eq. (1), $\lfloor \cdot \rfloor$, is the standard floor function which returns the integer portion of the provided value.

$$N_{m,t} = \left\lfloor \frac{\dot{\mathcal{V}}_m}{\frac{\pi}{6} d_m^3} \right\rfloor \quad and \quad N_t = \sum_m N_{m,t} \qquad (1)$$

The discrete feed rate values, $N_{m,t}$ and N_t, are also used to establish the distribution of incoming particles. The distribution is stored in an array where each element contains the index of a single mass phase (Fig. 2). The number of elements that contain the index for mass phase-m is proportional to the ratio $N_{m,t}/N_t$. When a particle is seeded, a random element of the distribution array is selected and the properties corresponding to the associated index are assigned to the new particle. By assigning mass phase values in this manner, the local distribution of incoming particles remains random while the globally set distribution is preserved.

The discrete feed rate is then adjusted to the solids time step, Δt_s, by determining the placement frequency of particles, F_p (Eq. 2). If F_p is equal to one, then one or more particles is placed every solids time step. If the value is greater than one, F_p represents the number of solids time steps that transpire between placements of individual particles.

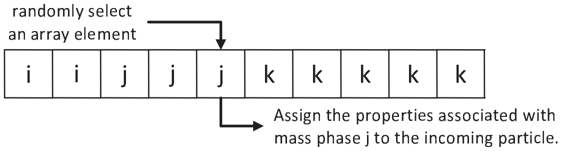
randomly select
an array element
i
i
j
j
j
k
k
k
k
k
Assign the properties associated with
mass phase j to the incoming particle.

Figure 2: Example distribution array with ten elements, three mass phase indices (i,j,k) with the respective distribution (20%,30%,50%). The fifth array element is randomly selected, leading to the incoming particle being assigned the physical properties associated with mass phase j.

$$F_p = \left\lfloor \frac{1}{\mathrm{N_t} \Delta t_s} \right\rfloor \qquad (2)$$

Once the placement frequency is known, an initial velocity for each incoming solids phase is determined. The feed rate and solids properties are combined with the cross-sectional inlet area, A, to calculate the corresponding initial velocities for each mass phase.

$$v_m = \frac{\dot{\mathcal{V}}_m}{\frac{\rho_{bm}}{\rho_m} * A} \qquad (3)$$

The phase-m velocities are then used to determine a uniform inlet velocity that is applied to all phases. This prevents particle overlaps from being generated within the staging area. Without this condition an overlap might arise due to any velocity differences, for example, if a fast moving particle becomes seeded behind a slower moving particle (Fig. 3). Once one of the particles fully enters the domain, the overlap would be treated as if it were generated by a collision. A sufficiently large overlap would create an artificial spike in the total kinetic energy, adversely affecting the integrity of the model.

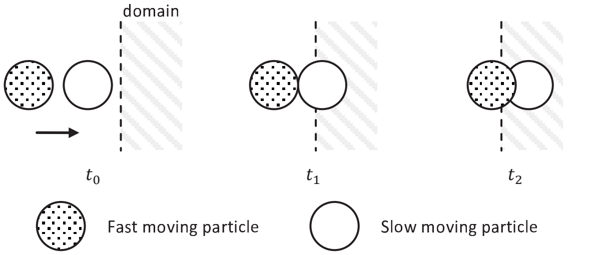

Figure 3: Motivation for a uniform inlet velocity. A slow moving particle is seeded behind a fast moving particle generating an artificial overlap.

The uniform inlet velocity is selected so that it minimizes the error between user-provided and calculated bulk density values. This is accomplished through a simple iterative procedure described below.

To begin the iterative selection process for the uniform inlet velocity, the minimum and maximum phase-m velocities specified by Eq. (3), $\min_m v_m$ and $\max_m v_m$, are used to generate $(s+1)$ test velocity values, v_i (Eq. 4). Here s is a parameter with integer value, like 100, that influences the number of iterations.

$$v_i \in \left\{ \min_m v_m + i * \frac{\max_m v_m - \min_m v_m}{s} \middle| i \in \{0,1,\dots,s\} \right\} \quad (4)$$

For a given test velocity, a corresponding test bulk density is computed for each mass phase, $\tilde{\rho}_{bm}(v_i)$. Here, A, is the area of the inlet region.

$$\tilde{\rho}_{bm}(v_i) = \frac{\rho_m \dot{V}_m}{v_i * A} \quad (5)$$

At this point the total error, Err_{v_i}, between the user-defined and newly computed test bulk density values is calculated. The test velocity that produces the least amount of total bulk density error is selected. This process is repeated with a new set of test velocities established around the minimizing velocity of the previous iteration, continuing until the difference between the results of two consecutive iterations is less than a specified tolerance (e.g., 10^{-4}).

$$Err_{v_i} = \sum_m |\rho_{bm} - \tilde{\rho}_{bm}(v_i)| \quad (6)$$

Finally, this optimized uniform inlet velocity is compared to two velocity values, explained below. The first defines a lower bound on the inlet velocity, while the second identifies the method used to seed incoming particles. These comparisons assure that the user-defined inlet specifications can be successfully managed and which of the two seeding algorithms, also presented below, is employed.

The lower bound on the inlet velocity is the minimum operational velocity, v_{min}. It coincides with a situation where particles are seeded in the staging area faster than they leave. Below v_{min} a new particle may be seeded such that it overlaps with an existing particle still within the staging area. Again, if the overlap is sufficiently large, it may cause an artificial spike in the kinetic energy of the system.

To prevent particle overlap within the staging area, a sub-grid is constructed to keep particles fully separated (Fig. 4). The number of cells in the sub-grid, N_p, depends on the dimensions of the boundary inlet, l and w, and the maximum particle diameter, $d_{max} = \max_m d_m$.

$$N_p = \left\lfloor \frac{l}{d_{max}} \right\rfloor \left\lfloor \frac{w}{d_{max}} \right\rfloor \quad (7)$$

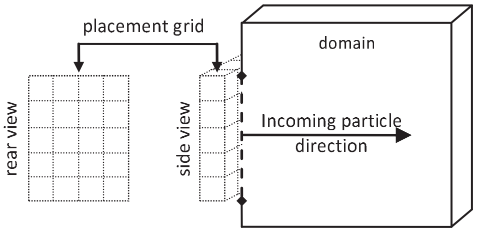

Figure 4: A grid structure is created within the staging area to ensure particles are not placed in contact with each other while being seeded.

This value is equivalent to the maximum number of disks of maximum diameter, d_{max}, that can be positioned in a square packing arrangement in a rectangle of the given dimensions. The cell sizes are uniform with their dimension based upon the maximum particle diameter to ensure that every cell can accommodate all specified particle diameters. This is a necessary condition because the diameter of an incoming particle is not established until the time of placement.

A minimum inlet velocity is then defined from the sub-grid and feed rate properties. Recall that if the placement frequency, F_p, is equal to one, then k particles are to be seeded every solids time step, and if F_p is greater than one, one particle ($k = 1$) is to be seeded every F_p solids time steps. Thus, with N_p locations available within the sub-grid, it follows that $\lfloor N_p/k \rfloor$ placements can occur before two particles are seeded in the same cell. Therefore, the first k particles that are seeded must exit the sub-grid by the time the $(\lfloor N_p/k \rfloor + 1)$ placement occurs. This assures that $(k + N_p - k\lfloor N_p/k \rfloor)$ cells are available to receive a new particle without it overlapping a previously seeded particle (Fig. 5). The minimum inlet velocity (Eq. 8) is the direct result of this requirement, establishing the lower bound on incoming particle velocity.

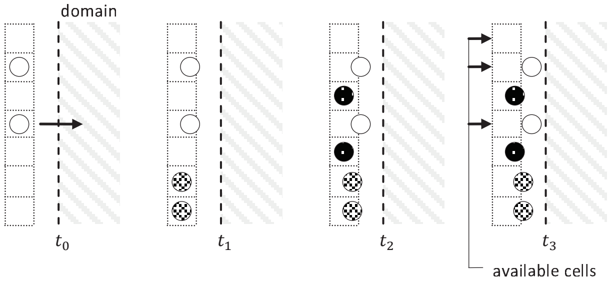

Figure 5: Illustration of the seeding process of the grid structure with $N_p = 7, F_p = 1, k = 2,$

$$v_{min} = \frac{d_{max}}{\Delta t_s F_p \left\lfloor \frac{N_p}{k} \right\rfloor} \quad (8)$$

Once the lower bound is defined, the manner in which particles are seeded into the staging area is determined. Particles can either be placed in an ordered arrangement or randomly distributed throughout the staging area.

If the staging area is densely populated, randomly placing particles has the potential to become computationally expensive. Therefore it is reserved for instances when a square packing would occupy less than half of the staging area at any given instant. The lower bound for the velocity correlating to less than half occupancy is given by Eq. (9).

$$v_{rand} = \frac{d_{max}}{\Delta t_s F_p \left\lfloor \left\lfloor \frac{N_p}{2} \right\rfloor / k \right\rfloor} \quad (9)$$

With the velocities v_{min} and v_{rand} indentified, the placement method of particles is determined. If the inlet velocity is between v_{min} and v_{rand}, particles are placed in the square packing arrangement defined by the grid structure imposed on the staging area. In this instance a fill order for the grid cells is generated. This pattern is cycled through when seeding particles.

For velocities greater than v_{rand} the particles are seeded via a guess-and-check (random) method. First a particle is assigned a random location within the staging area. This position is independent of the sub-grid imposed on the staging area. Then a check verifies if the location would cause a particle overlap. If there is an overlap, the particle is assigned a new random location within the staging area. This process continues until the particle has been placed without any overlaps.

After a particle is seeded into the staging area, the ghost cell that contains the new particle is determined (see Fig. 1). The cells within the computational domain that border those ghost cells in the inlet area are also identified. Finally, a local search is conducted that allows for particles already within the computational domain to identify incoming particles as possible contact partners.

Boundary Condition Operation

During the simulation, the placement frequency is used to calculate when the next particle is seeded. During each solids time step, the current time value is compared to a placement time, t_p. If $t_p$is less than or equal to the current solids time, a new particle is seeded into the inlet region, and the next seeding time updated.

$$t_p = t_p + F_p \Delta t_s \tag{10}$$

When a new particle is being seeded, a flag is set identifying it as a new particle which indicates the dismissal of any contact forces the particle is exposed to during entry. This classification is only maintained until the particle has fully entered the computational domain. As a result, entering (new) particles will always push obstructing particles in the computational domain away from the inlet.

Next, physical properties are assigned to the new particle through the appointment of a mass phase index. As previously discussed, an element from the array representing the incoming particle distribution is randomly selected. That array element contains the index of a single mass phase. From this selection, the physical properties associated with that index are assigned to the incoming particle.

After the mass phase is assigned, the center location of the new particle is determined. For an inlet using an ordered placement scheme, the sub-grid cell that will contain the new particle is identified. If the dimensions of the cell are larger than the diameter of the particle, the particle is randomly positioned within the cell; the particles that have delimited the cell dimensions are placed on cell-center. For an inlet using the random placement, the sub-grid imposed on the staging area is discarded and a guess-and-check process initiated. New locations within the staging area are assigned to the particle until it is placed without overlapping another particle.

Finally, the ghost cell that contains the new particle is determined. A local search is conducted allowing particles already within the computational domain to identify incoming particles as possible contact partners.

The positions of incoming particles are monitored during each solids time step. Once a particle has fully entered the computational domain, the flag identifying the particle as new is removed. This reclassification allows the particle to react appropriately to any force that acts upon it.

EXAMPLES

Provided below are two examples illustrating different applications of the DMIBC. The first example shows a 2D packed bed with a discrete mass inlet supplying new particles through the bottom of the bed. The second shows a 2D gas-solid system with an inlet gas jet coupled with a discrete mass inlet feeding particles into the bottom of the system. Both examples initially contain 500 particles comprised of five mass phases with a number distribution for each phase of 0.15, 0.20, 0.30, 0.20, 0.15. Table 1 lists the physical and operating parameters of the DMIBC used in both simulations.

Table 1: Particle and feed properties for both examples

Phase Index	d_m (cm)	ρ_m (g/cm^3)	$\dot{V}_m$ (cm^3/sec)	$\rho_{b,m}$ (g/cm^3)	Corresponding Distribution
1	0.100	1.75	0.00785	0.0125	0.15
2	0.125	1.85	0.02045	0.0325	0.20
3	0.150	1.95	0.05301	0.0950	0.30
4	0.175	2.05	0.05612	0.1000	0.20
5	0.200	2.15	0.06283	0.1500	0.15

In the first example, particles are fed into the bottom of a packed bed. The feed rate, N_t, is approximately 100 particles per second with a distribution matching the initial particle distribution. The computational domain is 3cm wide by 20cm tall.

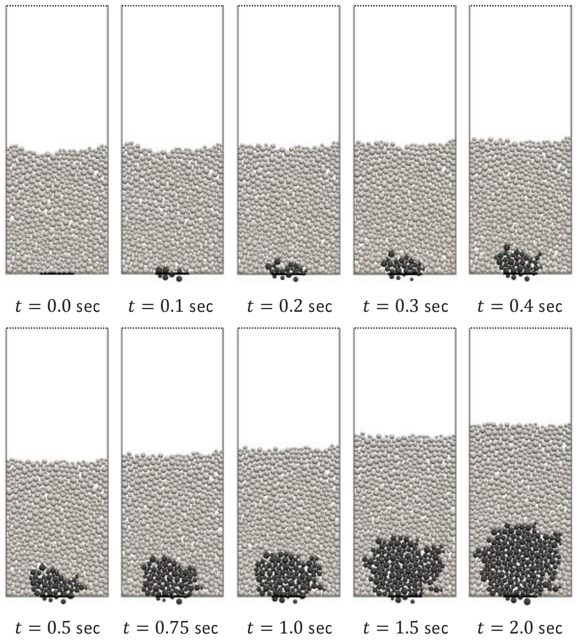

Figure 6: Time series (in seconds) of particle flow for a 2D example illustrating a DMIBC injecting into a packed particle bed.

Snapshots of the particle configuration within the system are shown in Figure 6, with only the lower 8cm displayed. For identification purposes, new particles have been shaded darker than the particles initially in the system. This series of images demonstrates that incoming particles push particles already in the system away from the inlet region as they enter the computational domain.

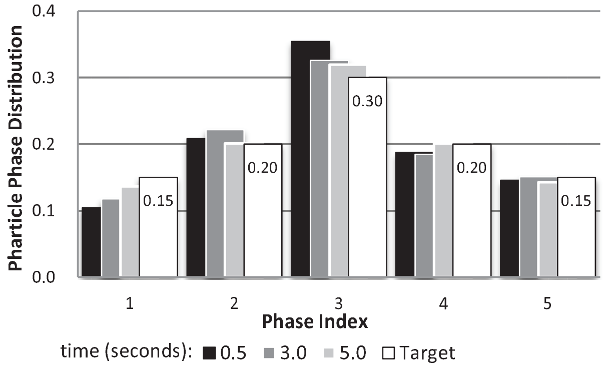

Figure 7: Cumulative incoming particle distributions for times t∈{0.5,3.0,5.0} during the simulation of a DMIBC into a packed bed.

Figure 7 shows the total incoming particle distributions at three times during the simulation. For each mass phase, the targeted proportion is shown to the right of the calculated values. As illustrated, the desired global distribution of particles is maintained approaching the target distribution as the simulation progresses through time.

The second example shows a discrete mass inlet coupled with a gas inflow boundary condition. In this simulation, the computational domain is 3cm wide and 50cm tall. A 1cm gas jet is centered on the bottom of the domain with a velocity of 1200cm/sec. A pressure outlet for the gas is specified along the top of the domain.

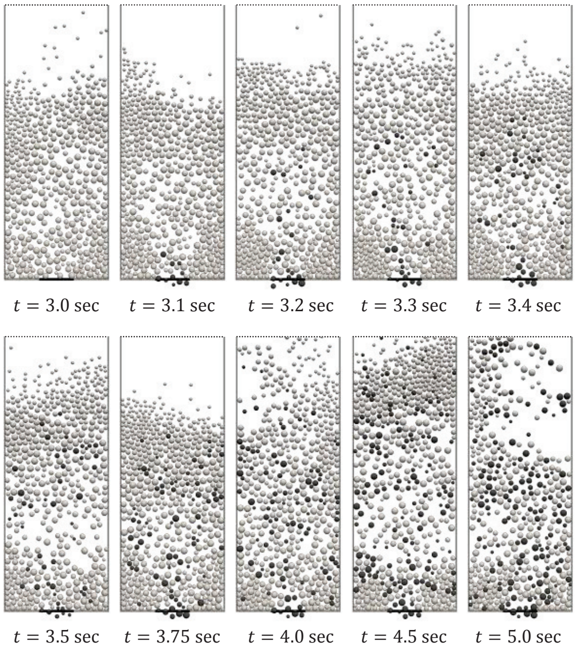

Figure 8: Time series (in seconds) of particle flow for a 2D example illustrating the coupling of a DMIBC with a gas jet.

The simulation starts with the initial 500 particles in a packed bed arrangement. During the first 3 seconds (not shown), a pseudo-steady-state is established between the gas flow and the initial particles during which no particles are fed into the system. After this initial transient, the particles are fed into the system at a rate of 100 particles per second with a distribution matching the initial distribution of particles within the system (see Table 1).

Snapshots of the particle configuration within the system are shown in Figure 8, with only the lower 8cm displayed. Again, to distinguish new particles from initial particles, the new particles have been shaded darker. Here, new particles are seen entering the computational domain and mixing with particles already in the system. This demonstrates that a DMIBC can be coupled with a continuum gas boundary condition for additional physical modeling possibilities.

The distribution of incoming particles (not shown) closely matches the results from the first example since both inlets are defined by the flow properties.

CONCLUSIONS

The current MFIX-DEM code has been modified allowing a more dynamic system by permitting particles to enter the computational domain during a simulation through user-defined boundary conditions. This feature can be used in simulations where particles enter as feedstock, like coal in a gasification model, or systems where particles are recycled, like flow to and from a downcomer in a circulating fluidized bed model.

ACKNOWLEDGEMENTS

This research was supported in part by an appointment to the National Energy Technology Laboratory Research Participation Program, sponsored by the U.S. Department of Energy and administered by the Oak Ridge Institute for Science and Education.

REFERENCES

[1] Boyalakuntla, D.S., “Simulation of Granular and Gas-Solid Flows Using Discrete Element Method.” Ph.D. diss., Carnegie Mellon University, 2003

[2] Cundall, P.A. and Strack, O.D.L. “A discrete element model for granular assemblies”. *Geotechnique*, Vol. 29, No. 1, pp. 47-65, 1979.

[3] Wang, F., Yu, Z., Marashdeh, Q., and Fan L.S. “Horizontal gas and gas/solid jet penetration in a gas-solid fluidized bed”. *Chem. Eng. Sci.*, Vol. 65, pp. 3394-3408, 2010

[4] Zhang, K., Zhang, J., and Zhang, B., “Experimental and numerical study of fluid dynamic parameters in a jetting fluidized bed of a binary mixture”, *Powder Technol.*, Vol. 132, pp. 30-38, 2003

[5] Garg, R., Galvin, J., Li, T., and Pannala, S., “Verification studies of open-source MFIX-DEM software for gas-solid flows,” submitted to *Power Technol.* March 29, 2010