

```

void do_simulation_step()
{
    double      const move_x      = 0.01;
    double      const move_y      = 0.01;
    unsigned int const my_object_label = 1u;

    glue::Phase const my_phase      = glue::make_phase(engine, my_object_label);

    std::vector<double> px;
    std::vector<double> py;

    glue::get_sub_range_current(engine, my_phase, px, py);

    std::vector<double> px_new (px.begin(), px.end());
    std::vector<double> py_new (py.begin(), py.end());

    for (unsigned int i = 0u; i != py_new.size(); ++i)
    {
        px_new[i] += move_x;
        py_new[i] += move_y;
    }

    glue::set_sub_range_target(engine, my_phase, px_new, py_new);
}

```

Step 1: Get a phase object

Step 2: Get current coordinates

Step 3: Set target coordinates
for phase

Current and target attributes are in-built named attributes...
Target attributes must be set for all vertices in phase. If no
target attributes are specified then it is implicitly assumed
target is equal to current attribute values.

Step 1: Create and clear a user custom field named phi

```
inline void setup_fields( grit::engine2d_type & engine )
{
    engine.attributes().create_attribute( "phi", 2u );
    glue::clear_attribute( engine, "phi", 0.0, glue::FACE_ATTRIBUTE() );
}
```

One can have custom attributes for vertices (0u), edges (1u) and faces (2u).

```
inline void do_simulation_step(
    grit::engine2d_type & engine
    , util::ConfigFile const & settings
)
{
    std::vector<double> phi; // Resulting potential
    glue::Phase const domain = glue::make_phase(engine);
    glue::get_sub_range(engine, domain, "phi", phi, glue::FACE_ATTRIBUTE() );
    //... Compute a new phi field value ...
    glue::set_sub_range(engine, domain, "phi", phi, glue::FACE_ATTRIBUTE() );
}
```

Step 2: Get the field values

Step 3: Set the field values

Use glue::VERTEX_ATTRIBUTE, glue::EDGE_ATTRIBUTE or glue::FACE_ATTRIBUTE to tell GLUE what kind of attributes you are working on in glue::clear_attribute, glue::copy_attribute, and glue::set_sub_range & glue::get_sub_range.