

1 Theory and Background

Reference : smith94c: The reaction ensemble method for the computer simulation of chemical and phase equilibria. . .

Definition of the reaction:

$$\sum_{i=1}^s \nu_i a_i = 0 \quad (1)$$

where $\{\nu_i\}$ is the set of stoichiometric coefficients ($\nu > 0$ for products and $\nu < 0$ for reactants), s is the number of species partaking in the reaction

a_i is the “formula vector” of species i and defines the molecular composition of species i .

Conservation of mass:

$$\sum_{i=1}^s a_{ji} N_i = B_j, \quad (2)$$

where B_j is the total number of atoms of type j in the system, constant at all times!

$$N_i = N_i^0 + \nu_i \xi, \quad (3)$$

where N_i^0 is the number of molecules of type i obeying eqn: 2 and ξ is the extent of the reaction for our purposes this is an integer.

$\xi > 0$: “forward” reaction

$\xi < 0$: “backward” reaction

example:



Forward step: choose randomly an entity of type A and B, and transform them to C and D

Backward step: choose randomly from C and D and back-transform to A and B

$$\bar{\nu} = \sum_{i=1}^s \nu_i, \quad (5)$$

$\bar{\nu}$ is the change of the total particle number per reaction

Calculating the transition probability between state k , prior to a reaction, and l , after a reaction step:

$$P_{kl}^\xi = \min \left(1, \prod_{i=1}^s \left[\left(\frac{V q_i}{\Lambda_i^3} \right)^{\nu_i \xi} \frac{(N_i^0)!}{(N_i^0 + \nu_i \xi)!} \right] \exp(-\beta \Delta U_{kl}) \right) \quad (6)$$

$$= \min \left(1, V^{\bar{\nu} \xi} \Gamma^\xi \prod_{i=1}^s \left[\frac{(N_i^0)!}{(N_i^0 + \nu_i \xi)!} \right] \exp(-\beta \Delta U_{kl}) \right), \quad (7)$$

where $\Gamma(T) = \prod_i \left(\frac{q_i}{\Lambda_i^3} \right)^{\nu_i} = \exp(-\Delta G^0/RT) (P^0/RT)^{\bar{\nu}}$, where $K = \exp(-\Delta G^0/RT)$ is the Dissociation constant. $\Delta U_{kl} = U_l - U_k$

Reaction: $HA \leftrightarrow H^+ + A^-$

Forward: $HA \rightarrow H^+ + A^-$

N_i is the number of particles of type i prior to the reaction

$$P_{\text{diss}} = \min \left(1, V\Gamma \frac{N_{HA}}{(N_{H^+} + 1)(N_{A^-} + 1)} \exp(-\beta\Delta U) \right) \quad (8)$$

Backwards: $H^+ + A^- \rightarrow HA$

$$P_{\text{back}} = \min \left(1, \frac{\Gamma_b}{V} \frac{N_{H^+}N_{A^-}}{N_{HA} + 1} \exp(-\beta\Delta U) \right), \quad (9)$$

where $\Gamma_b = \Gamma^{-1}$

2 Implementation

A reaction in the computer simulation may be viewed as a Monte Carlo trial move and is then implemented as such. In order for everything to work together, ESPRESSO is used to generate new trial conformations of the system. Then a sweep of reactions is tried, randomly choosing forward and backward reactions among all the different reactions, that are considered in the simulation. It is important to make sure the simulation obeys detailed balance! So the whole simulation is then a MD/MC hybrid.

Example implementation in tcl using the grandcanonical extension to the `part` command. Trial forward reaction: Acid dissociation

```
proc try_HA_diss {} {
    global type_HA type_A type_H val_HA val_H val_A K_HA_diss volume beta
    set E_old [expr [analyze energy total] - [analyze energy kinetic]]
    set N_HA [part gc number $type_HA]
    set N_A [part gc number $type_A]
    set N_H [part gc number $type_H]

    # "do" reaction change charge of HA particle
    set HA_id [part gc find $type_HA]
    if { $HA_id == -1 } {
        #no particle found abort, reaction failed
        return 0
    }
    part $HA_id q $val_A type $type_A

    set H_id [create_particle $type_H]
    integrate 0

    set E_new [expr [analyze energy total] - [analyze energy kinetic]]

    # calculate boltzmann factor
    set bf [expr $volume * $K_HA_diss * $N_HA / ( ($N_H + 1) * ( $N_A + 1 ) ) \
```

```

* exp( -$beta*( $E_new - $E_old ))]
if { [t_random] < $bf } {
    #accept
    return 1
} else {
    #reject
    delete_particle $H_id
    part $HA_id q $val_HA type $type_HA
    integrate 0
    return 0
}
}
}

```

A redirecting function that calls the chosen reaction and returns if the trial move was accepted or not:

```

proc try_reaction { reaction_id } {
    global type_H type_OH
    set succ 0
    if { $reaction_id == 0 } {
        # HA -> +H + -A
        # K_HA
        set succ [try_HA_diss]
    } elseif { $reaction_id == 1 } {
        # +H + -A -> HA
        # 1/K_HA
        set succ [try_HA_back]
    } elseif { $reaction_id == 2 } {
        # H2O -> +H + -OH
        # K_W 10^{-14}
        set succ [try_water_diss]
    } elseif { $reaction_id == 3 } {
        # +H + -OH -> H2O
        # 10^{14}
        set succ [try_water_back]
    } elseif { $reaction_id == 4 } {
        #add salt ions
        set succ [try_particle_insert]
    } elseif { $reaction_id == 5 } {
        #remove salt ion pair
        set succ [try_particle_delete]
    }
    return $succ
}
}

```

The main sampling routine of the script:

```

for {set cycle 0} {$cycle < $int_cycles} {incr cycle} {
    integrate $int_steps
}

```

```

for { set i 0 } { $i < $number_reactions } { incr i } {
    set reaction_id [expr int(floor([t_random]*$total_num_reactions))
    while { $reaction_id == $total_num_reactions } {
        # for the incredibly rare case that t_random returns a 1
        set reaction_id [expr int(floor([t_random]*$total_num_rea
    }
    incr reaction_counter($reaction_id)
    if { [try_reaction $reaction_id] } {
        incr reaction_acc($reaction_id)
    } else {
        incr reaction_rej($reaction_id)
    }
}
ANALYZE observables
.
.
.
}

```

1. Set up/warmup
2. Main cycle
 - MD run
 - MC sweep

3 Some results

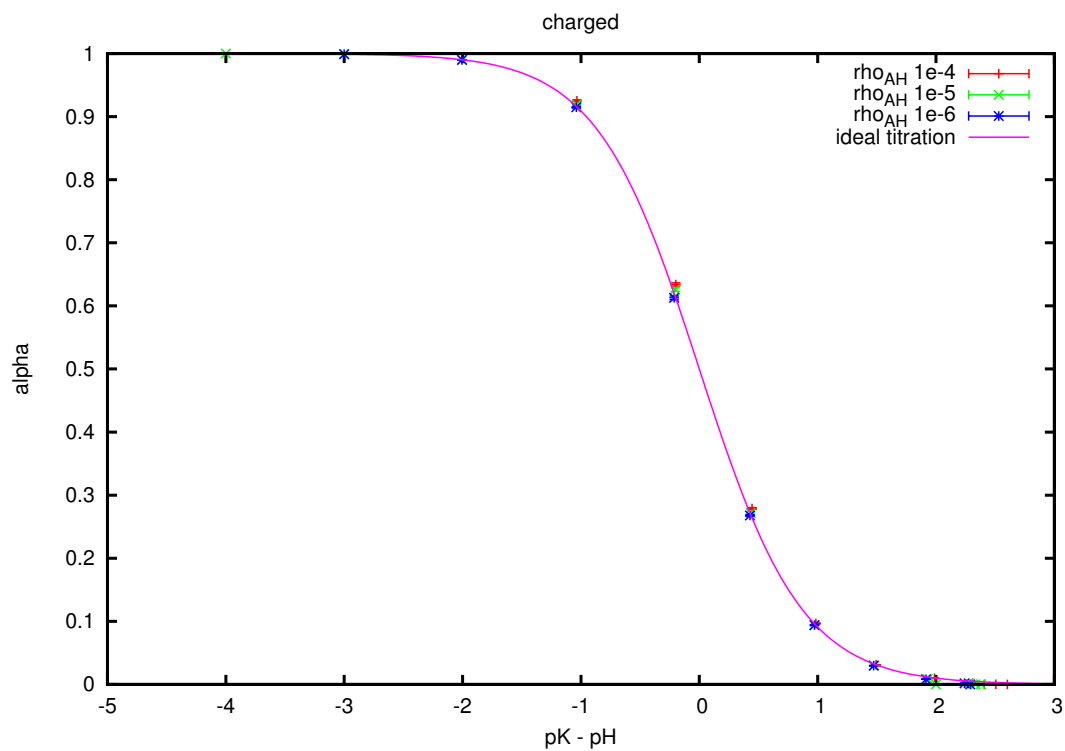


Figure 1: Titration curve of monomeric acid with 500 HA particles at different densities.

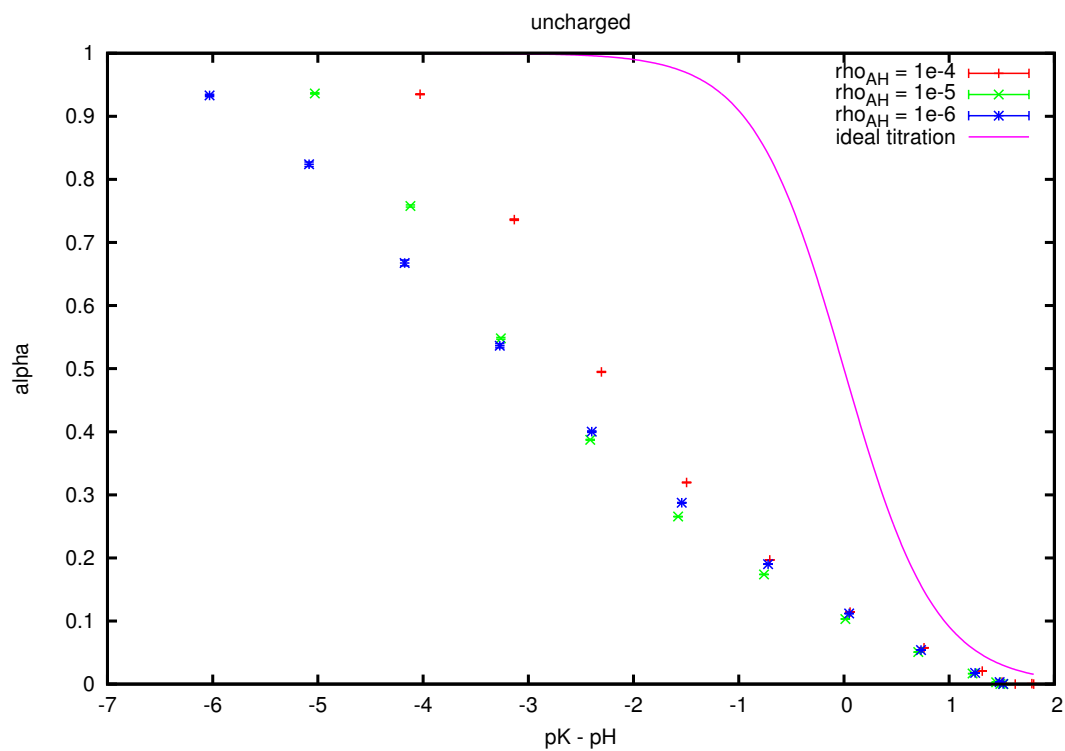


Figure 2: Titration curve for a single polyelectrolyte chain with 59 monomers.

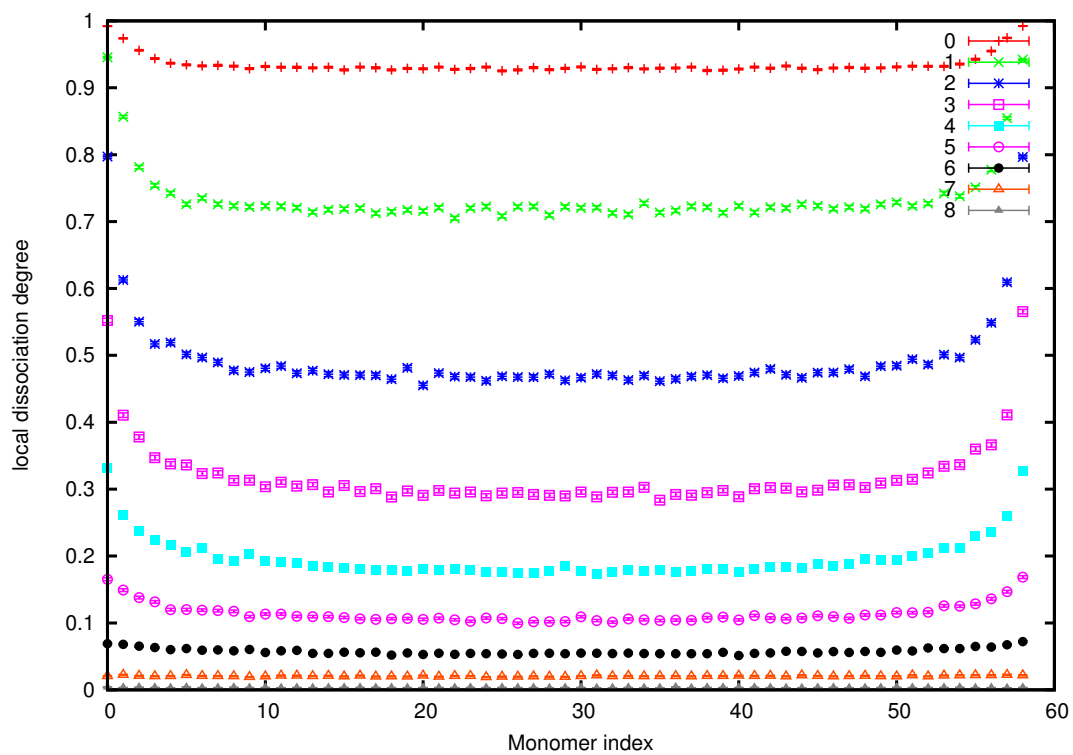


Figure 3: local dissociation degree of a polyacid at density 10^{-4} .

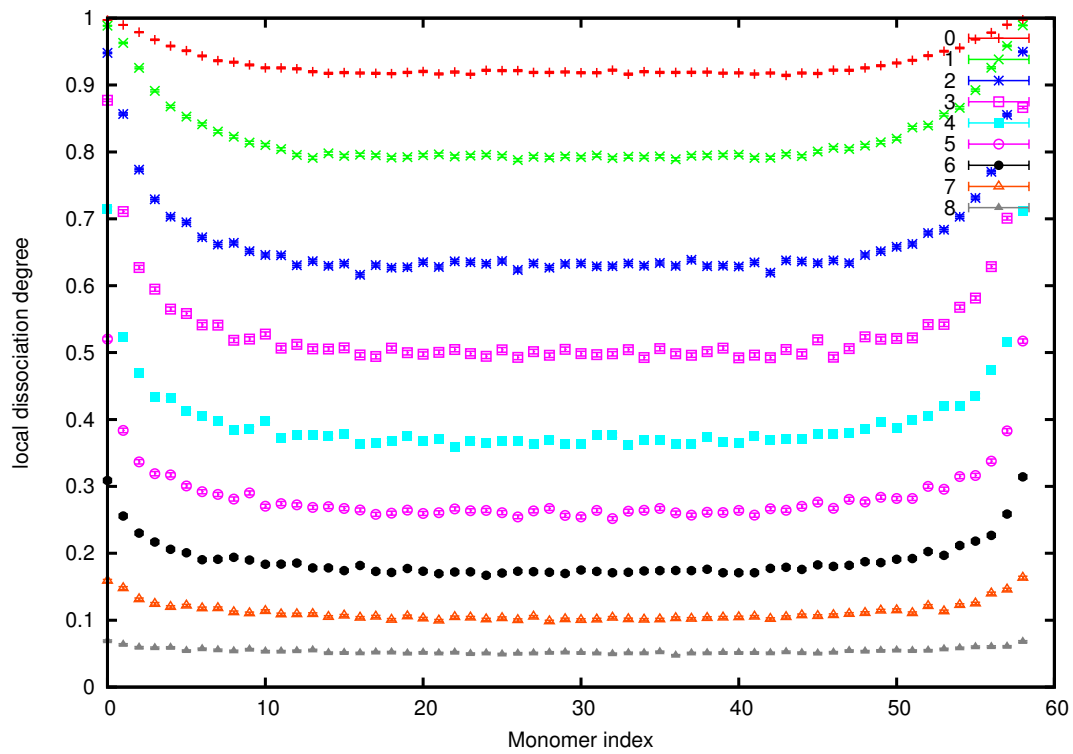


Figure 4: local dissociation degree of a polyacid at density 10^{-6} .

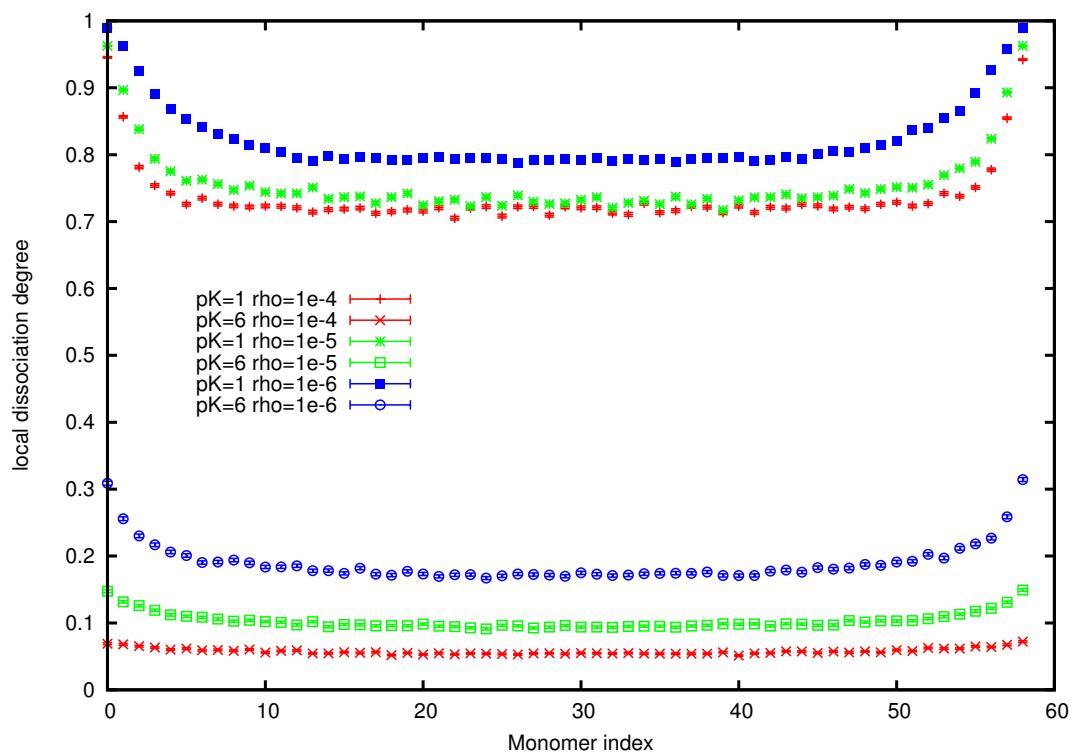


Figure 5: Dissociation profile at two pK values for different monomer densities.