The units of "rinv" and "sh_ewald" are inconsistent.

The following expression in "gromacs/gmxlib/nonbonded/nb_generic.cpp":

\[ rinvcorr = \begin{cases} 
\text{fr->coulomb Modifier == eintmodPOTSHIFT) ? rinv-fr->ic->sh_ewald : rinv;} 
\end{cases} \]

Unit of "rinv" is inverse of nanometer;
"sh_ewald" is just a number without unit in that
"ic->sh_ewald = gmx_erfc(ic->ewaldcoeff_q*ic->rcoulomb);"
is defined in
"gromacs/mdlib/forcerec.cpp"
or
"gromacs/ewald/pme-load-balancing.cpp"

Fixes #2215
Change-Id: Ia2ea57f3bd9d521879783b207353d9d6f4ccb4a8

There seems to be a minor, but embarrassing, issue here. sh_ewald is used as if it was computed as \( \text{erfc}(\beta \times \text{cutoff})/\text{cutoff} \) but computed only as \( \text{erfc}(\beta \times \text{cutoff}) \). Fortunately, cutoff is approximately one, and the errors from opposite charges approximately cancel, and for a given cutoff energy differences will have even more fortuitous cancelling.

I think we should subtract the cutoff distance from sh_ewald in forcerec.cpp. And plan to use a name that is more descriptive of the quantity, ie that it is a potential shift (albeit in 1/r form).

This issue was introduced along with the Verlet scheme, and now affects the group scheme also, in every Ewald kernel that I have seen.
Note that this bug only affects the reported energies, not the forces or the sampling. And, as Mark noted, the effects on the energy are very small, so this is relatively harmless (but still embarrassing).

#5 - 08/15/2017 04:35 PM - Mark Abraham
- Status changed from Fix uploaded to Resolved

Applied in changeset 3f970c00ffaf60b47c140003e72f941b429d275b.

#6 - 09/12/2017 11:58 AM - Mark Abraham
- Status changed from Resolved to Closed