Mem. S.A.It. Suppl. Vol. 1, 130

Memorie della Supplementi

Toward self-consistent models of Galaxy Evolution

Giovanni Carraro[1], Claudio Dalla Vecchia[2], and Bruno Jungwiert[3]

[1] Dipartimento di Astronomia, Università di Padova, vicolo dell'Osservatorio 2, I-35122, Padova, Italy e-mail: giovanni.carraro@unipd.it
[2] ICC - University od Durham, Ogden Center, South Road, Durham DH1 3LE - UK e-mail: Claudio.Dalla-Vecchia@durham.ac.uk
[3] Astronomical Institute, Academy of Sciences, Boční II 1401, 141 31 Praha 4, Czech Republic e-mail: bruno@ig.cas.cz

Abstract. After a brief description of the parallel N-body hydro-dynamical code we developed in the last years, we present what we believe to be a nice result of our simulations: the chemo-dynamical evolution of the *globular cluster* Omega Centauri. We shall finally outline future directions of research, in particular the implementation of a multiphase treatment of the Interstellar Medium (ISM).

Key words. Supercomputing – N-body simulations– Galaxy formation

1. Introduction

Galaxies and clusters of galaxies are believed to result from the gravitational instability of density fluctuations existing in the matter distribution of the primordial Universe. These fluctuations in the earliest phases grow linearly, and the evolution of the Universe in this linear or quasi-linear regime is generally studied by means of analytical tools. Galaxies and clusters however are clumpy and highly non-linear systems, and cannot be studied analytically; their formation and evolution must be modeled and followed numerically. The standard procedure is to consider the matter density distribution emerging from different cosmological models, and then to simulate the non-linear regime of structure formation using numerical simulations. The most widely accepted cosmological scenario is the hierarchical Cold Dark Matter (CDM) model, which is based on two ideas. The first one is that the Universe for most of its life has been dominated by an unknown kind of collision-less material, called Dark Matter (DM), and the second one is that the structures growth proceeds hierarchically, the less massive objects forming before the most massive ones, which are assembled through the merging of smaller more ancient entities. The so-called Dark era has been widely studied starting from the pioneering work of Davis et al (1985), who compared the DM spatial distribution emerging from N-body simulations, with the galaxies catalogues available at that time. When more powerful computers were available, the properties of individual galaxy halos have been studied in great detail, concluding that independently from the cosmological model, the primordial density fluctuations spectrum and the halo mass, all the halos show uniform universal properties in their matter distribution. However on galaxies scale the evolution is governed not only by the DM, but also by the gas, whose dynamics regulate on large scale the formation of grand design spiral arms and extended thin disks, and on smaller scale the formation of stars and the interaction between stars and the multiphase interstellar medium (ISM). To understand how galaxies formed and evolved it is therefore necessary to couple gravitational forces and hydro-dynamics.

In numerical astrophysics gravity can be computed using different methods. Tree codes (Barnes & Hut 1986) in particular reduce the scaling to $N \times logN$ putting the system in a hierarchical structure, and computing the forces from distant particle groups in an approximate fashion.

Hydro-dynamical forces are calculated adopting a Lagrangian or Eulerian formalism. The use of grids is natural for Eulerian codes. On the other hand, Lagrangian codes start from the hydro-dynamical conservation laws in Lagrangian formalism, and mostly utilize particles to map the fluid properties. This is the case of the Smoothed Particle Hydro-dynamics (SPH) technique developed by Lucy (1977) and Gingold & Monaghan (1977). The advantage of this technique is the great flexibility and adaptivity (Hernquist & Katz 1989; Steinmetz & Müller 1993; Carraro et al. 1998).

2. The parallel TreeSPH code

Carraro et al. (1998) developed a pure particle code, combining Barnes & Hut (1986) octo-tree with SPH, and applying this code to the formation of a spiral galaxy like the Milky Way. The code is similar to Hernquist & Katz (1989) TreeSPH. It uses SPH to solve the hydro-dynamical equations. In SPH a fluid is sampled using particles, there is no resolution limitation due to the absence of grids, and great flexibility thanks to the use of a time and space dependent smoothing length. Shocks are captured by adopting an artificial viscosity tensor, and the neighbours search is performed using the octo-tree. The octo-tree, combined with SPH, allows a global time scaling of $N \times logN$. A good advantage of such codes is that it is easy to introduce new physics, like cooling and radiative processes, magnetic fields and so forth.

Galaxy formation requires a huge dynamical range (Davé et al. 1997). In fact an ideal galaxy formation simulation would start from a volume as large as the Universe to follow the initial growth of the cosmic structures, and at the same time would be able to resolve regions as small as Giant Molecular Cloud (GMC), where stars form and drive the galaxy evolution through their interaction with ISM. This ideal simulation would encompass a dynamic range of 10^9 (from Gpc to parsec), 10^6 time greater than that achievable with present day codes.

Big efforts have been made in the last years to enlarge as much as possible the dynamical range of numerical simulations, mainly using more and more powerful supercomputers: scalar and vector computers indeed cannot handle efficiently a number of particles greater than half a million (Katz et al. 1996).

A successful example is the Virgo Consortium (Glanz 1998), which has been recently able to perform simulations of the Hubble Volume (a cube of $2000h^{-1}Mpc$ on a side on the Cray T3E supercomputer by using a number of particles close to 10^9. They used a parallelized $P^3M - SPH$ code.

Davé et al. (1997) for the first time developed a parallel implementation of a TreeSPH code (PTreeSPH) which can follow both collision-less and collisional matter. They report results of simulations run on a Cray T3D computer of the adiabatic collapse of an initially isothermal gas sphere (using 4096 particles), of the

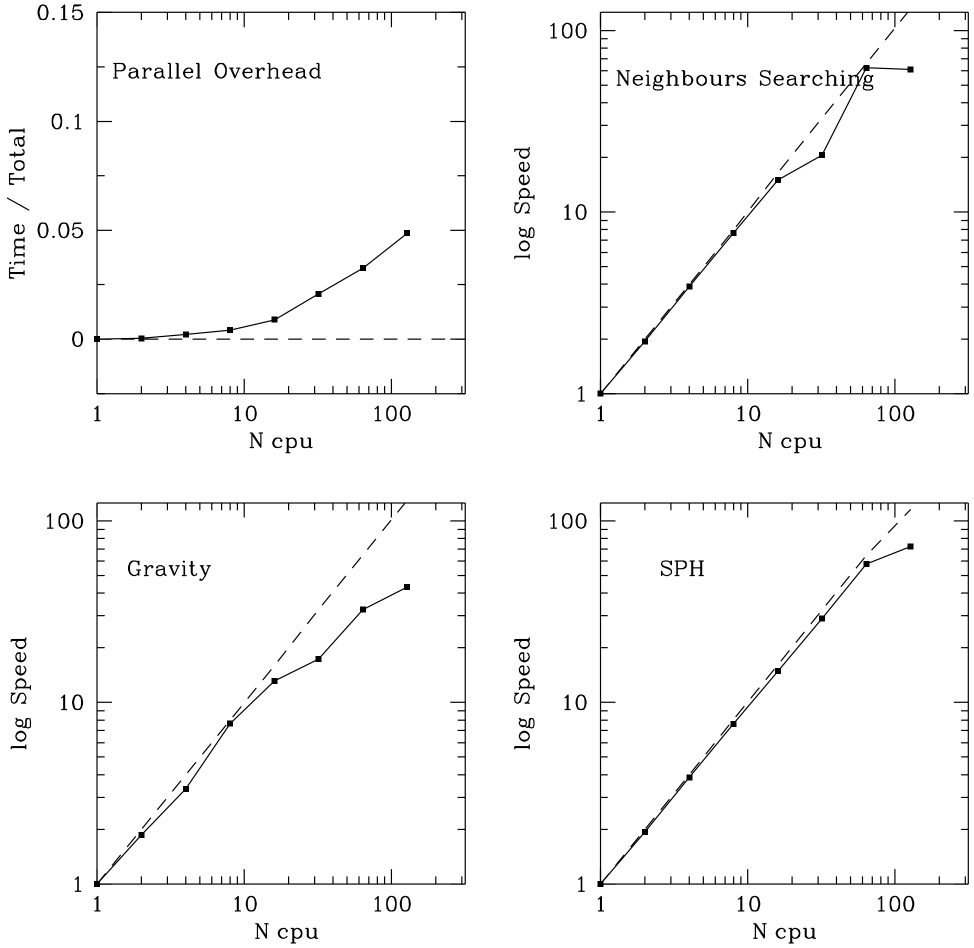

Fig. 1. Scalability of different code sections (solid lines), as reported in each panel, averaged on 50 time-steps. Dashed lines indicate the ideal scalability.

collapse of a Zel'dovich pancake (32768 particles) and of a cosmological simulation (32768 gas and 32768 dark particles).
Their results are rather encouraging, being quite similar to those obtained with the scalar TreeSPH code (Hernquist & Katz 1989). Porting a scalar code to a parallel machine is far from being an easy task. A massively parallel computer (like the Silicon Graphics T3E) links together hundreds or thousands of processors aiming at increasing significantly the computational power. For this reason they are very attractive, although several difficulties can arise in adapting a code to these machines. Any processor possesses its own memory, and can access other processors memory by means of communications which are handled by a hard-ware network, and are usually slower than the computational speed. Great attention must be paid to avoid situations in which a small number

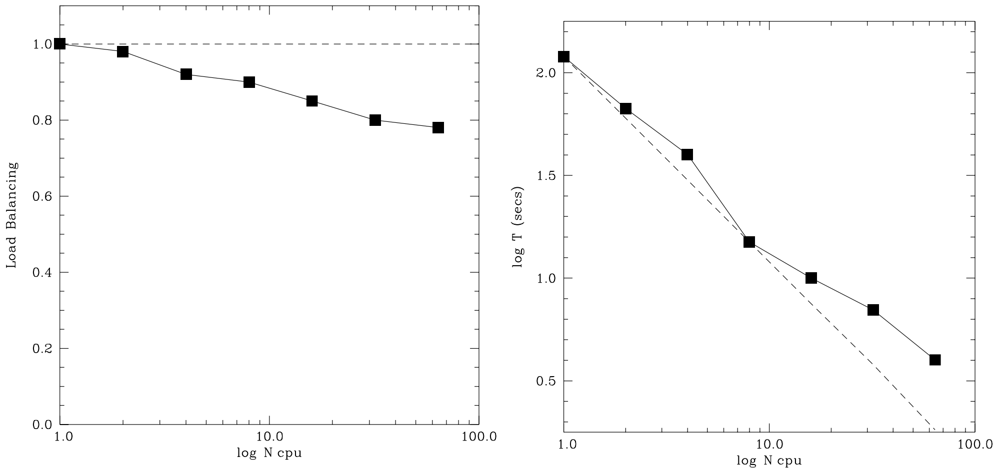

Fig. 2. Left Panel: overall code load-balance, averaged on 50 time-steps (solid line). Dashed line indicates ideal load-balance. Right Panel: code scalability in a pure collisionless simulation. See text for more details.

of processors are actually working while most of them are standing idle. Usually one has to invent a proper data distribution scheme which allows to subdivide particles into processors in a way that any processor handles about the same number of particles and does not need to make heavy communications. Moreover the computational load must be shared between processors, ensuring that processors exchange informations all together, in a synchronous way, or that any processor is performing different kinds of work when it is waiting for information coming from other processors, in an asynchronous fashion (Davé et al. 1997).

Lia & Carraro (2000) present a parallel implementation of the TreeSPH code described in Carraro et al. (1998). The numerical ingredients are the same as in the scalar version of the code. However the design of the parallel implementations required several changes to the original code. The key idea that guided us in building the parallel code was to avoid continuous communications, limiting the information exchange at a precise moment along the code flow. This clearly reduces the communication overhead. To handle communications we used the MPI libraries, which allow us to run the code in a variety of platforms (cluster of PC or workstations, IBM SP3 and SP4, Cray T3E, and Origin 2000 and 3000)(Dalla Vecchia 2001).

2.1. Benchmark

To evaluate the code performances, we use the adiabatic collapse simulation (Lia & Carraro 2000, Dalla Vecchia 2001) and perform simulations at increasing number of processors. We believe that this test is very stringent, and can give a lower limit of the code performances due to the high density contrast that is present at the time of maximum compression, when the particles are highly clustered. We are going to check the code timing, overall load-balance and scalability. Moreover we shall analyze in details particular sections of the code, like the gravity computations, the SPH and the neighbor searching. An estimate of the parallel over-head will be given as well.

2.2. *Timing analysis*

We run the adiabatic collapse test up to the time of the maximum compression (t $\simeq 1.1$) using 2×10^4 particles on 1, 2, 4, 8, 16, 32 and 64 processors, and looked at the performances in the following code sections (see also Table 1):

- total wall-clock time;
- data up-dating;
- parallel computation, which consists of barriers, the construction of the *ghost-tree* and the distribution of data between processors;
- search for neighbor particles;
- evaluation of the hydro-dynamical quantities;
- evaluation of the gravitational forces;
- miscellaneous, which encompasses I/O and statistics.

The results summarized in Table 1 present the total wall-clock time per processor over the last 50 time-steps, together with the time spent in each of the 5 subroutines (data updating, neighbor searching, SPH computation, gravitational interaction and parallel computation). The gravitation interaction takes about one-third of the total time, while the search for neighbors takes roughly comparable time. The evaluation of hydrodynamical quantities takes about one-fourth of the time, the remaining time being divided between I/O and data updating. The parallel over-head does not appear to be a serious problem, being at maximum about 1% of the total time. This timing refers, as indicated above, to simulations stopped at roughly the time of maximum compression. A run with 8 processors up to $t \simeq 2.5$, the time at which the system is almost completely virialized, took 3800 secs. Global code performances are analyzed in the next sections.

2.3. *Load-balance*

One of the most stringent requirements for a parallel code is the capability to distribute the computational work equally between all processors. This can be done defining a suitable work-load criterion (Lia & Carraro 2000). This is far from being an easy task (Davé et al. 1997), and in practice some processors stand idly for some time waiting that the processors with the heaviest computational load accomplish their work. This is true also when an asynchronous communication scheme is adopted, as in our TreeSPH code. We are using individual work-loads, based on the time spent to evaluate the gravitational interaction on one particle with all the other ones. A better choice would be to define the work-load only for active particles, which are the particles evolving fastly. This possibility is currently under investigation, due to the memory problems that can arise. To evaluate the code load-balance we adopted the same strategy of Davé et al. (1997), measuring the fractional amount of time spent idle in a time-step while another processor performs computation:

$$L = \frac{1}{N_{procs}} \sum_{j=1}^{N_{procs}} 1 - \frac{(t_{max - t_i})}{t_{max}}. \qquad (1)$$

Here t_{max} is the time spent by the slowest processor, while t_i is the time taken by the $i-th$ processors to perform computation. The results are shown in Fig. 2 (left panel), where we plot the load-balance for simulations at increasing number of processors, from 1 to 64. The load balance maintains always above 80%, being close to 1 up to 8 processors. For the kind of simulations we are performing, the use of 8 processors is particularly advantageous for symmetry reasons.

2.4. *Scalability*

At increasing number of processors, a parallel code should ideally speed up linearly. In practice the increase of the processors number causes an increase of the communications between processors, and a degradation of the code performances. To test this, we used the same simulations

Table 1. The Adiabatic Collapse test. Benchmarks for a run with 2×10^4 particles. Time refers to 50 time-stpes.

N_{cpu}	Total	Data Up-date	Parallel Over-head	Neighbors	SPH	Gravity	Miscellaneous
	secs	secs	secs	secs	secs	secs	secs
1	120	0.47	0.00	40	36	40	3.53
2	69	0.22	0.60	23	19	25	1.18
4	42	0.27	1.70	14	9.5	15	1.53
8	23	0.13	3.20	5.5	5.4	5.3	3.47
16	17.3	0.13	3.40	3.4	3.0	3.8	3.60
32	11.5	0.09	3.00	2.6	1.3	3.2	1.31
64	7.5	0.05	2.90	0.33	1.1	1.9	1.23

discussed above, running the adiabatic collapse test with 2×10^4 particles at increasing processors number. We estimated how the code speed scales computing the wall-clock time per processor spent to execute a single time-step, averaged over 50 time-steps. In Fig. 1 we plot the speed (in sec^{-1}) against the number of processors.
The code scalability keeps very close to the ideal scalability up to 8 processors, where it shows a minimum. This case in fact is the most symmetric one Then the scalability deviates significantly only when using more that 16 processors. Looking also at Fig. 1, it is easy to recognize that mainly the gravitational interaction is responsible for this deviation.

To better judge the code performances, we run a simulation of the collapse of a pure DM system, aiming at showing the scalability of the gravity section of the code. The results are shown in Fig. 2 (right panel). They are good up to 16 processors, afterwards they suddenly get worse. This trend does not change by introducing all the other code parts, as it will be shown in the next sections. This is clearly imputable to the dominant role of the gravity, which represents not only the most time consuming section of any TreeSPH code (this holds also for the serial code), but also by definition the less parallel part of the code.

3. The puzzling globular cluster Omega Centauri

ω Centauri is one of the most interesting globular cluster in the Milky Way, and maybe the most studied one (Majewski et al. 1999). The most striking feature of this cluster is the measured metallicity spread, which has been interpreted as the evidence of a multiple stellar population inside it (Norris et. al 1996, Suntzeff & Kraft 1996). The precise metallicity distribution function (MDF) is nonetheless still disputed. Norris et al. (1996) suggest the presence of a secondary peak at $[Ca/H] \approx -0.9$ roughly 5 times smaller that the main peak at $[Ca/H] \approx -1.4$. This trend is not confirmed by Suntzeff & Kraft (1996), who claim for a more regular MDF.
On the base of a large photometric survey Lee et al. (1999) analyzed the color distribution of a sample of bright stars, showing that on the average it has an e-folding trend, with the presence of several significant metallicity peaks. However Majewski et al. (1999) arrive at a somewhat different conclusion, showing that the MDF has a gaussian shape with the maximum at $[Fe/H] \approx -1.7$, and with some evidences of a secondary peak. Although different, all these analyses point to the common picture of an object which experienced an irregular self-enrichment over its evolution.
Putting together the chemical and kinematical properties Majewski et al. (1999) claim that ω Centauri might be a possible dwarf galaxy relict.

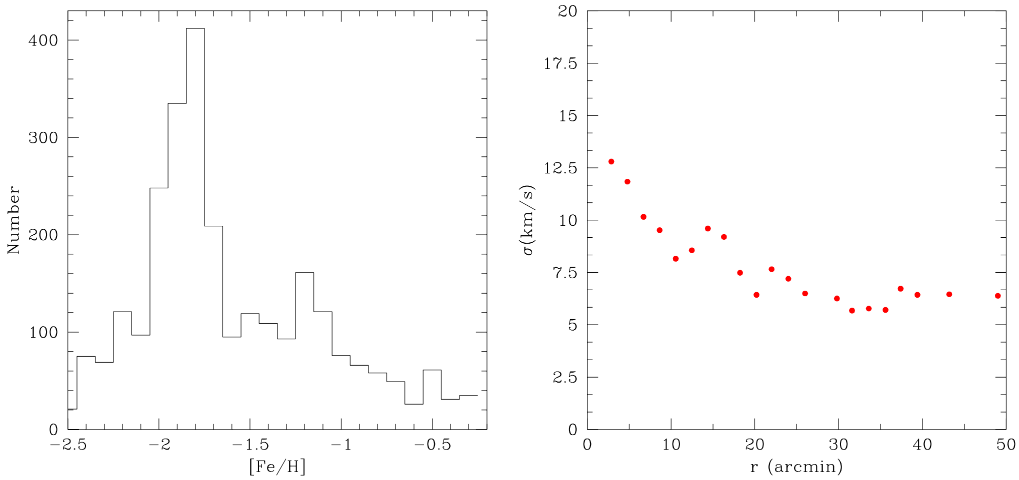

Fig. 3. Left panel: metallicity distribution function of simulated stars in Omega Centauri. Right panel: trend of radial velocity dispersion as a function of the distance from the center in the simulation of Omega Centauri.

Carraro & Lia (2000) propose a N-body/gasdynamical model for the formation and evolution of ω Centauri, suggesting that this globular cluster can actually be the remnant of a dwarf elliptical galaxy, formed and evolved avoiding strong mergers, and finally digested by the Milky Way.

3.1. Metallicity distribution

The stars MDF at the end of the simulation is presented in Fig. 3 (left panel). The bulk of stars has a metallicity in the range between $[Fe/H] = -1.9$ and $[Fe/H] - 1.6$. A secondary peak is observed at $[Fe/H] = -1.2$, about 4 times smaller than the major peak.
Finally a significant number of stars has a relatively high metallicity around $[Fe/H] = -0.50$. The global shape resembles an e-folding SF history (Carraro & Lia 2000), with secondary peaks which mark successive bursts of star formation.
The good agreement we find with the data presented by Majewski et al. (1999) confirms their suggestion that ω Centauri experienced an irregular self-enrichment over its evolution and may actually be the core of a larger dwarf elliptical galaxy.

3.2. Internal kinematics

The internal kinematics of ω Centauri has been analyzed in great detail by Merritt et al. (1997) using the radial velocity catalogue obtained with CORAVEL by Mayor et al. (1997). From this study it emerges that the cluster has a peak rotational speed of about 8 km/sec at 11 pc from the center. Moreover the cluster has a central velocity dispersion $\sigma = 17$ km/sec, the higher one among globular clusters.
From the analysis of our data we find evidences of smaller rotational speed (about 4 km/sec at 15 pc), whereas the velocity dispersion profile $\sigma(r)$ (see Fig. 3, right panel) shows a central value of about 13 km/sec, denoting that our model is somewhat colder than ω Centauri.

3.3. Dark Matter around ω Centauri?

Our simulation starts with a virialised DM halo, in whose center gas concentrates due

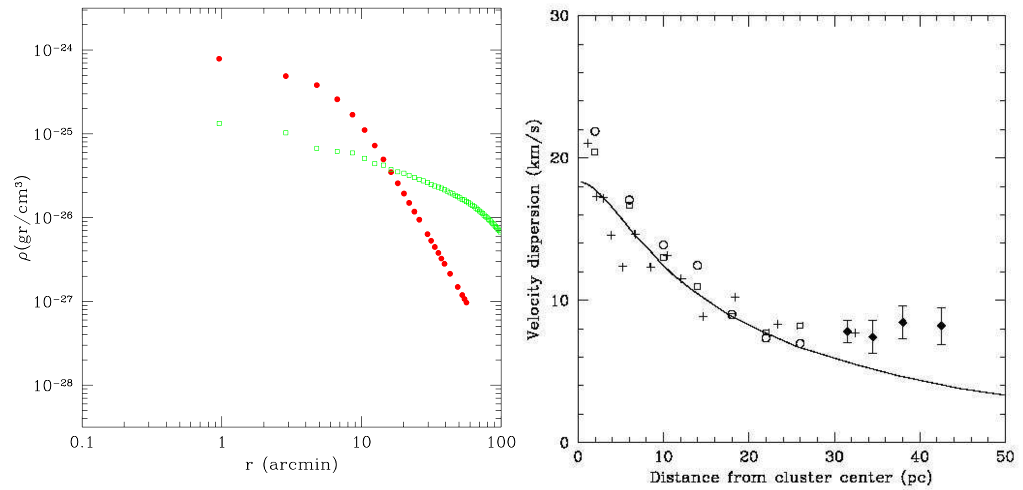

Fig. 4. Left panel: density profile of stars(filled symbols) and DM (open symbols) at the end of the simulation. Right panel: observations from Scarpa et al. (2002) of the radial velocity dispersion in the outskirts of Omega Centauri.This picture is a courtesy of Riccardo Scarpa.

to cooling instability, forming stars. The final stars and DM distribution is shown in Fig. 4 (left panel), where open squares refer to DM, and filled circles represent stars. The inner region of the cluster is dominated by stars up to 20 arcmin, which we call the transition radius r_{tr}. In the outer region DM dominates over a distance 6 times larger than the transition radius.
We expect that the internal kinematics of stars in the inner region is unlikely to be influenced by DM. On the contrary, stars outside the transition radius r_{tr} (20 arcmin) should show a kinematics strongly influenced by DM. A detail spectroscopic study of the stars kinematics in the cluster envelope should be able to confirm or deny the presence of DM. There is indeed in our model a trend of the velocity dispersion to weakly increase out of the transition radius. This result has been recently confirmed observationally by Scarpa et al. (2002), who measured the radial velocity dispersion of the stars in Omega Centauri halo (see Fig 4, right panel.)

4. A multiphase description of the Interstellar Medium

Gas crucially conditions star formation and evolution of large-scale instabilities (e.g. bars, spiral arms). In turn, star formation and large-scale flows induced by these instabilities influence mass and chemical profiles of galaxies (e.g. Martinet & Friedli 1997; Portinari & Chiosi 2000). The gas content in a given area of a galaxy is determined by the competing processes of star formation and stellar mass-loss and by spatial flows of gas. A coherent picture of galactic evolution must therefore consistently couple stellar and gaseous dynamics with star formation and stellar evolution. Star formation is, in large-scale models of galaxies, typically implemented through simplistic parametrizations reflecting mainly the energy supply from young stars and the gas mass fraction locked in newly born stars. On the other hand, stellar mass-loss has received much less attention from galactic N-body modellers since it was for a long time considered as a secondary issue. But nowadays, both observations and stellar evolutionary models indicate that the gas mass fraction restituted by stars is huge and may reach, when integrated over the stellar mass spectrum, some 45% over the Hubble time. Stars thus represent not only a place for permanent gas blocking but also an important temporary reservoir of gas that will be gradually reinjected into the interstellar medium. Recent computer models of galactic dynamics usually approximate the stellar mass-loss as instantaneous, i.e. happening at the moment of stellar birth. Nonetheless, stars lose matter during all their lives. Stellar lifetimes span a very large range and in the case of low-mass stars they compare with or overpass the Hubble time. The use of the instantaneous recycling approximation, which was proposed by Tinsley (1980) for high mass-stars, is therefore not satisfactory for the whole stellar mass spectrum. From the above emerges an obvious motivation for building a computer code able to follow the galactic structure and dynamics together with a non-instantaneous gas recycling. One can expect that the connection of such a recycling with the dynamics will affect large-scale gravitational instabilities, spatial flows of matter, star formation and gas consumption rates, etc., and thus the long-term evolution of galaxies.

Jungwiert et al. (2001) and Lia et al. (2002) developed implementations of the gas recycling in their N-body codes. But, while the former implementation lacks the treatment of the chemical evolution, the latter lacks the description of the cold phase of the ISM.

Therefore we start a project to include a new kind of particles in our TreeSPH code, which are able to follow the cold phase of the ISM. In details, while the warm phase of the ISM is described by SPH particles, the cold phase is described by Sticky Particles (Schwarz 1981). These particles are the places where Star Formation occurs, and, when some criteria are met, transform into stars. Stars evolve according to Stellar Evolution rules, and deliver energy, gas and metals to the ISM. The released gas, which is hot, is described by SPH particle. Due to cooling, SPH particles can cool and transform into Sticky Particles, closing the loop. A nice example of this scheme is in Semelin & Combes (2002), although this scheme lacks a reliable treatment of the Chemical Evolution. We are presently in a coding phase, and we would like to apply the new code to the study of the chemical evolution of the Galactic Disk, in particular to the development of the Galactic chemical abundance gradient. Some related computations are presented in Jungwiert et al. (2003).

5. Conclusions

We have summarized the numerical work done by our group in the last few years, and presented an application to the *globular cluster* Omega Centauri.

Finally, we have outlined future perspective, e.g. the study of the chemodynamical evolution of galactic disks by using a multiphase treatment of the ISM. Work is in progress to complete this implementation and compare simulations with data from nearby stars, HII region and galactic open clusters.

Acknowledgements. We would like to deeply thanks CINECA and CNAA for the time allocated for this project. G. Carraro expresses his gratitude to Prof. Carlos Frenk for his hospitality in Durham, and Profs. Petr Harmanec and Jan Palouš for their hospitality in Prague. This research is partly supported by the grant *No. 202/01/D075* of the Grant Agency of the Czech Republic and by Padova University through the *International Relationship* program.

References

Barnes J.E., Hut P., 1986, Nature 324, 446
Carraro G., Lia C., Chiosi C., 1998, MNRAS 297, 1029
Carraro G., Lia C., 2000, A&A 357, 977
Dalla Vecchia C., 2001, Master Thesis, Padova University

Davé R., Dubinski J., Hernquist L.E., 1997, New Astronomy 2, 277
Davis M., Efstathiou G., Frenk C.S., White S.D.M., 1985, ApJ 292, 371
Gingold R.A., Monaghan J.J., 1977, MNRAS 181, 375
Glanz J., 1998, Science 280, 1522
Jungwiert B., Combes F., Palouš J., 2001, A&A 276, 85
Jungwiert B., Carraro G., Dalla Vecchia C., 2003, *Jenam 2002* Proceedings, in press
Hernquist L.E., Katz N., 1989, ApJS 70, 419
Katz N., Weinberg D.H., Hernquist L.E., 1996, ApJS 105, 19
Lee Y.-W, Joo J.-M et al., 1999, Nature 402, 55
Lia C., Carraro G., 2000, MNRAS 314, 145
Lia C., Portinari L., Carraro G., 2002, MNRAS 330, 821
Lucy L., 1977, AJ 82, 1013
Majewski S.R. et al., 1999, in *The galactic Halo: from Globular Clusters to Field stars*, ASP Conf. Ser 189, 457
Martinet, L., & Friedli, D., 1997, A&A, 323, 363
Mayor M., Meylan G. et al., 1997, AJ 114, 1087
Merritt D., Meylan G., Mayor M., 1997, AJ 114, 1074
Norris J.E., Freeman K.S., Mighell J., 1996, ApJ 462, 251
Portinari, L., & Chiosi, C., 2000, A&A 355, 929
Scarpa R., Marconi G., Gilmozzi R., 2002, New Astronomy, in press
Semelin B., Combes F., 2002, A&A, 338, 826
Steinmetz M., Muller E., 1993, A&A 268, 391
Suntzeff N.B., Kraft R.P., 1996, AJ 111, 1913
Schwarz P. 1981, ApJ 247, 77
Tinsley, B.M., 1980, Fund. Cosmic Phys. 5, 287