SemiCore Code Workshop

Damon H. T. Cheung, Jacky H. T. Yip

September 2, 2020

Contact information

Department of Physics
The Chinese University of Hong Kong

Damon H. T. Cheung

- 1155127032@link.cuhk.edu.hk

Jacky H. T. Yip

- jacky.ht.yip@link.cuhk.edu.hk
- github.com/jhtyip

Let us know if you have questions, ideas, want to discuss or have discovered any bugs!

Physics behind SemiCore code: Decaying Dark Matter

$$\psi^* \to \psi + l$$

where ψ^* = mother (mom), ψ = daughter (dau),
l = relativistic light particle $\Rightarrow$ escaped

Parameters: V_k = recoil velocity of ψ, τ = half life

Since $\dfrac{V_k}{c} \approx \dfrac{\Delta m}{m_{\psi^*}} \Rightarrow m_{mom} \approx m_{dau}$ and

$$\langle \boldsymbol{v_{dau}} \rangle = \langle \boldsymbol{v_{mom}} + V_k \hat{\boldsymbol{n}} \rangle \Rightarrow \langle \boldsymbol{v_{dau}} \rangle = \langle \boldsymbol{v_{mom}} \rangle$$

($\hat{\boldsymbol{n}}$ is random), energies and angular momenta can be related by:

$$E_{k,dau} = E_{k,mom} + \frac{1}{2} {V_k}^2$$
$$\boldsymbol{L_{dau}} = \boldsymbol{L_{mom}}$$

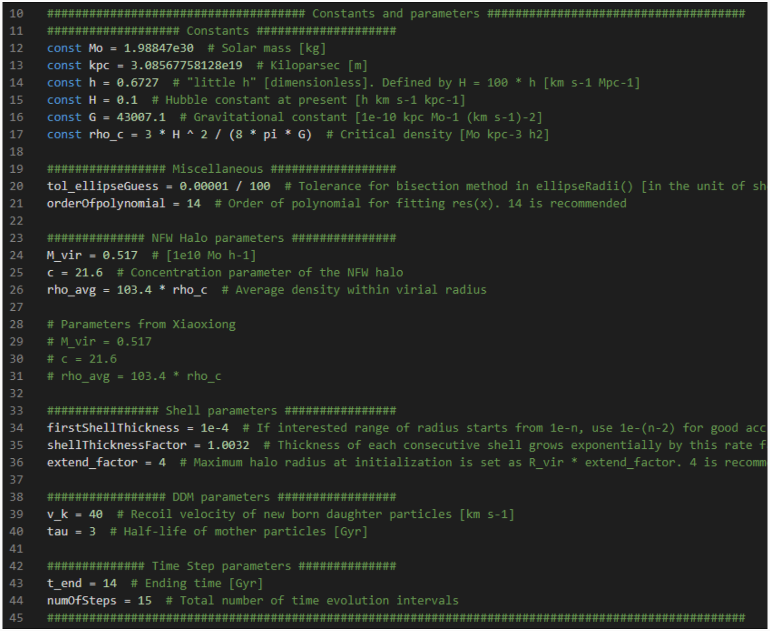
General information

2 scripts written in Julia (1.4.2): **main.jl** (parameters & algorithm), **myFunctions.jl** (functions used by main.jl)

- No package required
- Physics parameters: $\tau,\ V_k,\ M_{vir},\ c,\ \Delta_{vir}$[1]
- Default units for constants and parameters:
  - Dimensionless "little h"[2]: $h = 0.6727$
  - Mass: $10^{10} M_\odot\ h^{-1}$
  - Length: $kpc\, h^{-1}$
  - Velocity: $km\ s^{-1}$
- GPE, K.E., L, etc are in terms of quantity **per unit mass**

[1] $\rho_{avg, r=r_{vir}} = \Delta_{vir} \rho_c$
[2] $H_0 = 100\ h\ kms^{-1}\ Mpc^{-1}$

Where to adjust constants and parameters: main.jl



```julia
##################################### Constants and parameters #####################################
################### Constants ###################
const Mo = 1.98847e30  # Solar mass [kg]
const kpc = 3.08567758128e19  # Kiloparsec [m]
const h = 0.6727  # "little h" [dimensionless]. Defined by H = 100 * h [km s-1 Mpc-1]
const H = 0.1  # Hubble constant at present [h km s-1 kpc-1]
const G = 43007.1  # Gravitational constant [1e-10 kpc Mo-1 (km s-1)-2]
const rho_c = 3 * H ^ 2 / (8 * pi * G)  # Critical density [Mo kpc-3 h2]

################# Miscellaneous ##################
tol_ellipseGuess = 0.00001 / 100  # Tolerance for bisection method in ellipseRadii() [in the unit of sh
orderOfpolynomial = 14  # Order of polynomial for fitting res(x). 14 is recommended

############## NFW Halo parameters ###############
M_vir = 0.517  # [1e10 Mo h-1]
c = 21.6  # Concentration parameter of the NFW halo
rho_avg = 103.4 * rho_c  # Average density within virial radius

# Parameters from Xiaoxiong
# M_vir = 0.517
# c = 21.6
# rho_avg = 103.4 * rho_c

################ Shell parameters ################
firstShellThickness = 1e-4  # If interested range of radius starts from 1e-n, use 1e-(n-2) for good acc
shellThicknessFactor = 1.0032  # Thickness of each consecutive shell grows exponentially by this rate f
extend_factor = 4  # Maximum halo radius at initialization is set as R_vir * extend_factor. 4 is recomm

################# DDM parameters #################
v_k = 40  # Recoil velocity of new born daughter particles [km s-1]
tau = 3  # Half-life of mother particles [Gyr]

############## Time Step parameters ##############
t_end = 14  # Ending time [Gyr]
numOfSteps = 15  # Total number of time evolution intervals
####################################################################################################
```

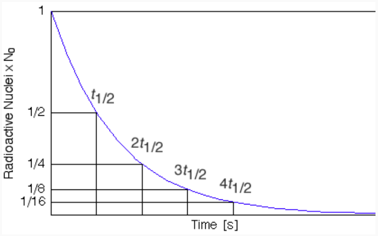
Time discretization

Mass m of mothers decays as: $m_{mom} = m_0 e^{-ln2\frac{t}{\tau}}$

Previous method: constant $\Delta t = \frac{t\_end}{numOfSteps} \left(= \frac{Age\ of\ universe}{No.\ of\ time\ steps}\right)$

Now: set Δt for each step such that $\Delta m_{mom} = constant$

- More computationally cost-effective: calculate more frequently in the early phase during which more mass decays



Previous method: equal intervals in time
(physicsabout.com)

Shell scheme

Spherical symmetry is assumed. The dark matter halo is partitioned into spherical shells, one enclosing another

Therefore, the main objects being manipulated in the code are **arrays of shells** named **X**$shells\_$**content**:

- Particle type **X**: M (mom), D (dau), T (mom + dau)
- **content**: $radii$, $mass$, $enclosedMass$, GPE, L, $totalE$, etc

The ith item in any of such arrays corresponds to the **content** for that particle type **X** in the ith shell. For example:

- $Dshells\_mass[i]$: mass of daus in the ith shell
- $Mshells\_L[i]$: Angular momentum of moms in the ith shell

Initialization: $Tshells\_radii$

At initialization, fixed positions and thicknesses of all shells are defined. $Tshells\_radii$ stores all info specifying each shell and **remains unchanged throughout the running of the code**:

- $Tshells\_radii[i,\ 1]$: inner radius of the ith shell
- $Tshells\_radii[i,\ 2]$: outer radius
- $Tshells\_radii[i,\ 3]$: shell radius = (inner + outer) / 2. It is generally used to represent the position of the ith shell when e.g. calculating GPE

For efficiency, thickness (outer - inner) of the ith shell is $\alpha\beta^{i-1}$

- $\alpha = firstShellThickness$ and $\beta = shellThicknessFactor$
- Adjust α and β for more or fewer $numOfShells$

Initialization: $Tshells\_radii$

All together, the shells span the halo space from $r = 0$ (center) to $r = 4R_{vir}$

$Tshells\_radii$ **being unchanged means that masses falling outside of $4R_{vir}$ are neglected**

- Discrepancy is negligible

Why $4R_{vir}$ (or larger; tunable via $extend\_factor$) but not R_{vir}?

- To trace mass contributions to the halo's center from more daughter particles
- More accurate GPE calculation

Initialization: $Mshells\_mass$, $Dshells\_mass$, $Tshells\_mass$

Initializing the $Xshells\_mass$ arrays:

- NFW mass profile (analytic) $\Rightarrow Mshell\_mass[i]$
- No daughter at initialization $\Rightarrow Dshell\_mass[i] = 0$
- $Tshells\_mass[i] = Mshells\_mass[i] + Dshells\_mass[i]$

Together with $Tshells\_radii$, all useful quantities can be inferred

Therefore, the job of the code is to calculate the right amount of masses of moms and daus in each shell at the end of each decay time step

GPE: $Tshells\_GPE$

At **initialization**:

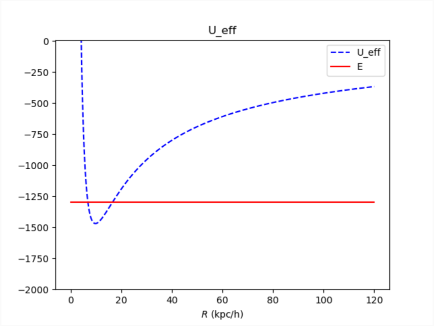
- $\Phi_{NFW}(r)$ is used (analytic)

In **other time steps**:

- $\Phi(r) = -G\ \left[\frac{1}{r}\int_0^r dM(r') + \int_r^R \frac{dM(r')}{r'}\right]$
- Numerically, $\int \rightarrow \sum$ and $dM(r') \rightarrow Tshells\_mass[i]$
- Shell radii are used, i.e. $r' = Tshells\_radii[i,\ 3]$
- Mass in each shell is thought to concentrate at just beyond the shell radius
- Φ values at all shell radii are calculated and stored
  - $\Phi(r)$ at arbitrary r can be further obtained by interpolating these values (required in next step)

Roots finding: r_{max}, r_{min} of a daughter's orbit

Assume that every mother orbits circularly at the bottom of the effective potential $\Phi_{eff}(r) = \Phi(r) + \frac{L^2}{2r^2}$



Solve $\Phi_{eff}(r) = E_{k,dau}$ for r_{max}, r_{min} using bisection method

- Increase accuracy by lowering error tolerance $tol\_ellipseGuess$
- Escaped daughters ($E_{k,dau} \geq 0$) are discarded

g-function: mass contribution from daughters

Recall: goal is to obtain resulting mass in each shell

- Calculate daughters' contribution to each shell via **g-function**

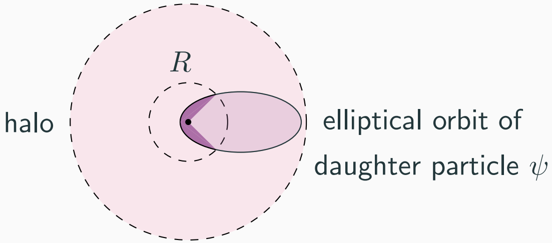
g-function: mass contribution from daughters

Let ψ be a daughter particle in an elliptical orbit and $M(R)$ is its mass contribution to the region enclosed by $r = R$

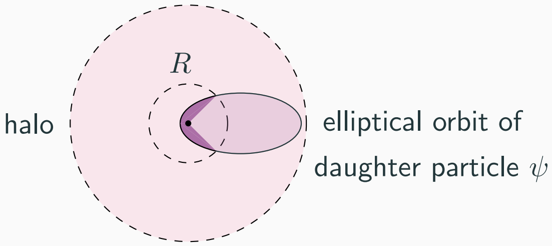
$$M(R) = \sum m_\psi g_\psi(R)$$

The **g-function** $g_\psi(R)$ is the ratio defined by $\Delta t_\psi(R)$ [the duration in which ψ orbits within R] to T_ψ [ψ's orbital period]:

$$g_\psi(R) = \Delta t_\psi(R)/T_\psi$$



g-function: mass contribution from daughters



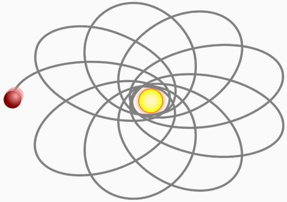
$$g_\psi(R) = \Delta t_\psi(R)/T_\psi$$

- $r_{max} \leq R \Rightarrow$ entire ellipse inside $R \Rightarrow g_\psi(R) = 1$
- $r_{min} > R \Rightarrow$ entire ellipse outside $R \Rightarrow g_\psi(R) = 0$
- $r_{min} \leq R < r_{max} \Rightarrow g_\psi(R) = \Delta t_\psi(R)/T_\psi$

g-function: the accurate picture and its general formula

However, daughters do NOT travel in closed ellipses

Not a point mass field $\Rightarrow$ not exactly a conic section


(Wikipedia)

$g_\psi(R)$ must be evaluated via this **general formula**:

$$\Delta t_\psi(R)/T_\psi = \int_{r_{min}}^{R} \frac{dr}{\sqrt{E-\Phi_{eff}(r)}} \,/\, \int_{r_{min}}^{r_{max}} \frac{dr}{\sqrt{E-\Phi_{eff}(r)}}$$

g-function: integration method

Define the normalization: $x = \frac{r - r_{min}}{r_{max} - r_{min}}$

Modify the integrand $\frac{1}{\sqrt{E - \Phi_{eff}(r)}} =$
$\frac{1}{\sqrt{E-\Phi_{eff}(r)}} + \frac{1}{\sqrt{\Phi'_{eff}(x_{max})(x_{max}-x)}} + \frac{1}{\sqrt{\Phi'_{eff}(x_{min})(x-x_{min})}} -$
$\frac{1}{\sqrt{\Phi'_{eff}(x_{max})(x_{max}-x)}} - \frac{1}{\sqrt{\Phi'_{eff}(x_{min})(x-x_{min})}}$

Define $res(x) =$
$\frac{1}{\sqrt{E-\Phi_{eff}(r)}} - \frac{1}{\sqrt{\Phi'_{eff}(x_{max})(x_{max}-x)}} - \frac{1}{\sqrt{\Phi'_{eff}(x_{min})(x-x_{min})}}$

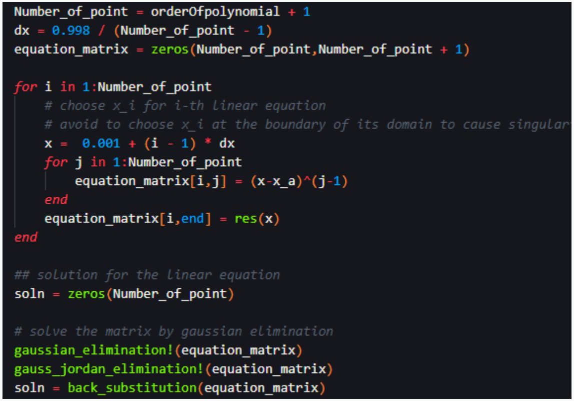
Employ **polynomial approximation**:
$res(x) = a + bx + cx^2 + dx^3 + ... + C_{N-1}x^{N-1}$

How: take N points to set up a system of linear equations and solve for coefficients by Gaussian elimination (order tunable via $orderOfpolynomial$)

g-function: code for polynomial approximation of $res(x)$

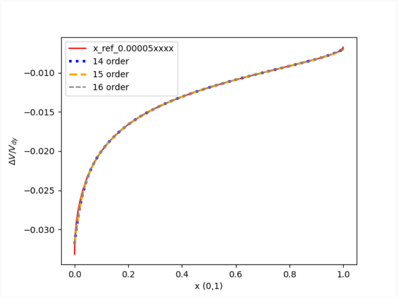
$N =$ number of points $= orderOfpolynomial + 1$

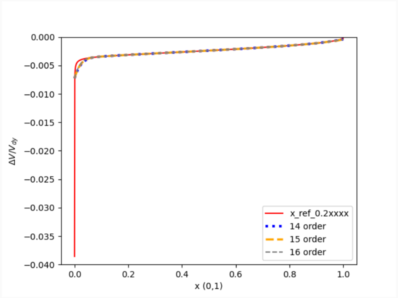
Use the set of points $x = [x_1, x_2, ..., x_N]$
where $x_1 = 0.001,\ x_2 = \frac{0.998}{N-1},\ x_3 = \frac{2(0.998)}{N-1}, ...,\ x_N = 0.999$



```
Number_of_point = orderOfpolynomial + 1
dx = 0.998 / (Number_of_point - 1)
equation_matrix = zeros(Number_of_point,Number_of_point + 1)

for i in 1:Number_of_point
    # choose x_i for i-th linear equation
    # avoid to choose x_i at the boundary of its domain to cause singular
    x =  0.001 + (i - 1) * dx
    for j in 1:Number_of_point
        equation_matrix[i,j] = (x-x_a)^(j-1)
    end
    equation_matrix[i,end] = res(x)
end

## solution for the linear equation
soln = zeros(Number_of_point)

# solve the matrix by gaussian elimination
gaussian_elimination!(equation_matrix)
gauss_jordan_elimination!(equation_matrix)
soln = back_substitution(equation_matrix)
```

g-function: plots of res(x) and polynomial approximations





- 14-th order polynomial is smooth and good enough
- Higher order (e.g. 21-st order) approximations give fluctuating curves, not recommended

g-function: integration method

Recall: modified integrand

$$\frac{1}{\sqrt{E - \Phi_{eff}(r)}} = res(x) - \frac{1}{\sqrt{\Phi'_{eff}(x_{max})(x_{max} - x)}} - \frac{1}{\sqrt{\Phi'_{eff}(x_{min})(x - x_{min})}}$$

All 3 terms can now be directly integrated. In particular:

$$\int res(x)dx = ax + bx^2/2 + cx^3/3 + dx^4/4 + ... + C_{N-1}x^N/N$$

Adiabatic expansion

After distributing the mass of newborn daughters, the halo's enclosed mass $M(R)$ changes from $M_{t_{i-1}}$ to M_{t_i}. Consider a test particle orbiting circularly at R and the conservation of its angular momentum L:

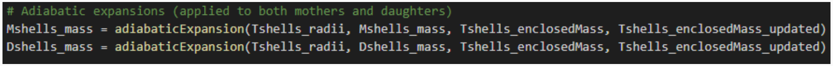
$$v^2 = \frac{GM}{R},\ L = mRv \Rightarrow \frac{L^2}{Gm^2} = MR$$

$$\Rightarrow M_{t_i} R_{t_i} = M_{t_{i-1}} R_{t_{i-1}} \Rightarrow R_{t_i} = \frac{M_{t_{i-1}}(R_{t_{i-1}})}{M_{t_i}(R_{t_{i-1}})} R_{t_{i-1}}$$

i.e. orbiting radius of the test particle evolves from $R_{t_{i-1}}$ to R_{t_i}

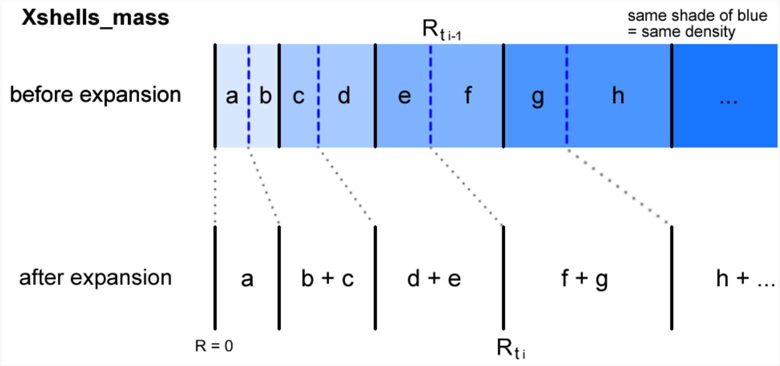
This **adiabatic expansion** of orbits is applied to both mothers ($Mshells\_mass$) and daughters ($Dshells\_mass$ which includes new daughters just born in the current time step)

Adiabatic expansion: implementation



```
# Adiabatic expansions (applied to both mothers and daughters)
Mshells_mass = adiabaticExpansion(Tshells_radii, Mshells_mass, Tshells_enclosedMass, Tshells_enclosedMass_updated)
Dshells_mass = adiabaticExpansion(Tshells_radii, Dshells_mass, Tshells_enclosedMass, Tshells_enclosedMass_updated)
```

What $adiabaticExpansion()$ does:



- Positions and dimensions of shells stay the same (boundaries in black)
- Given R_{t_i}, $R_{t_{i-1}}$ is retrieved by interpolation
- Assuming uniform density within each shell, obtain mass to be moved

Adiabatic expansion: more details on implementation

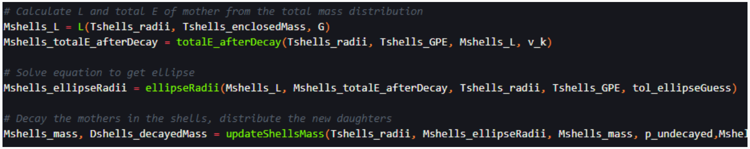
A more detailed explanation:

1. Recall: positions and dimensions of shells ($Tshells\_radii$) stay the same before and after expansion (boundaries in black). Goal is to figure out the amount of mass to be put in each shell after expansion
2. Using the set of boundaries (outer radii) as $\{R_{t_{i-1}}\}$, we use the relation $R_{t_i} = \frac{M_{t_{i-1}}(R_{t_{i-1}})}{M_{t_i}(R_{t_{i-1}})} R_{t_{i-1}}$ to compute the corresponding set of expanded radii $\{R_{t_i}\}$
3. Given an outer radius R_{t_i} **of the expanded halo**, retrieve the corresponding $R_{t_{i-1}}$ from the 2 sets of radii by interpolation (note: $R = 0$ does not expand)
4. Assuming uniform density within each shell before expansion, we can deduce the right parts of mass to be put in each shell after expansion

Summary: algorithm of dark matter only SemiCore code

- Initialize NFW profile (mothers only + NO daughters)
- Calculate GPE
- **For loop starts (t = 0)**
  - Mothers decay into daughters, solve for r_{max} and r_{min}
  - Calculate g-function to distribute daughters
  - Obtain updated enclosed mass profile
  - Adiabatic expansion of mothers and daughters
  - Update GPE
- **For loop ends (t = t_end)**

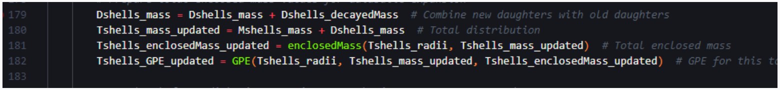
Code explained: for loop



```
# Calculate L and total E of mother from the total mass distribution
Mshells_L = L(Tshells_radii, Tshells_enclosedMass, G)
Mshells_totalE_afterDecay = totalE_afterDecay(Tshells_radii, Tshells_GPE, Mshells_L, v_k)

# Solve equation to get ellipse
Mshells_ellipseRadii = ellipseRadii(Mshells_L, Mshells_totalE_afterDecay, Tshells_radii, Tshells_GPE, tol_ellipseGuess)

# Decay the mothers in the shells, distribute the new daughters
Mshells_mass, Dshells_decayedMass = updateShellsMass(Tshells_radii, Mshells_ellipseRadii, Mshells_mass, p_undecayed,Mshel
```

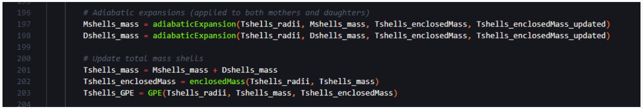
1. Obtain angular momenta L of mothers
2. Calculate total energy of newborn daughters by
   $$E_{k,dau} = E_{k,mom} + \frac{1}{2} m_{dau} {V_k}^2$$
3. Solve for r_{max} and r_{min}
4. Decaying of mothers and distribution of daughters
   - $Mshells\_mass$: decay by the ratio $= e^{\frac{-ln(2)}{\tau} t_i} / e^{\frac{-ln(2)}{\tau} t_{i-1}}$
   - $Dshells\_decayedMass$: distributed newborn daughters using g-function

Code explained: for loop



```
Dshells_mass = Dshells_mass + Dshells_decayedMass  # Combine new daughters with old daughters
Tshells_mass_updated = Mshells_mass + Dshells_mass  # Total distribution
Tshells_enclosedMass_updated = enclosedMass(Tshells_radii, Tshells_mass_updated)  # Total enclosed mass
Tshells_GPE_updated = GPE(Tshells_radii, Tshells_mass_updated, Tshells_enclosedMass_updated)  # GPE for this t
```

1. Add up accumulated (from previous steps) and newborn daus
2. Add up mothers and daughters
3. Obtain updated enclosed mass array
4. Obtain GPE (for record keeping, not useful)



```
# Adiabatic expansions (applied to both mothers and daughters)
Mshells_mass = adiabaticExpansion(Tshells_radii, Mshells_mass, Tshells_enclosedMass, Tshells_enclosedMass_updated)
Dshells_mass = adiabaticExpansion(Tshells_radii, Dshells_mass, Tshells_enclosedMass, Tshells_enclosedMass_updated)

# Update total mass shells
Tshells_mass = Mshells_mass + Dshells_mass
Tshells_enclosedMass = enclosedMass(Tshells_radii, Tshells_mass)
Tshells_GPE = GPE(Tshells_radii, Tshells_mass, Tshells_enclosedMass)
```

1. Adiabatic expansion of mothers
2. Adiabatic expansion of daughters
3. Add up mothers and daughters
4. Obtain updated enclosed mass array
5. Obtain updated GPE to be used in next loop

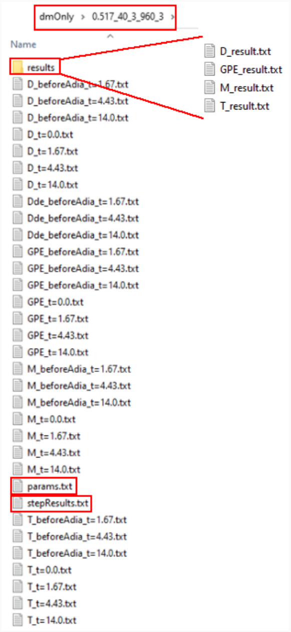
Output

The code creates a **folder** ("dmOnly") under the same directory as itself, and a **subfolder** named according to the parameters of that particular run ("$M_{vir}\_V_k\_\tau\_numOfShells\_numOfSteps$")

Inside the **subfolder**:

1. "params.txt" stores **ALL** constants and parameters necessary for reproducing the same result
2. "stepResults.txt" stores time used in each time step
3. Various files from each time step labeled accordingly
4. A folder ("results") containing result files from the end step

Output


dmOnly › 0.517_40_3_960_3 ›

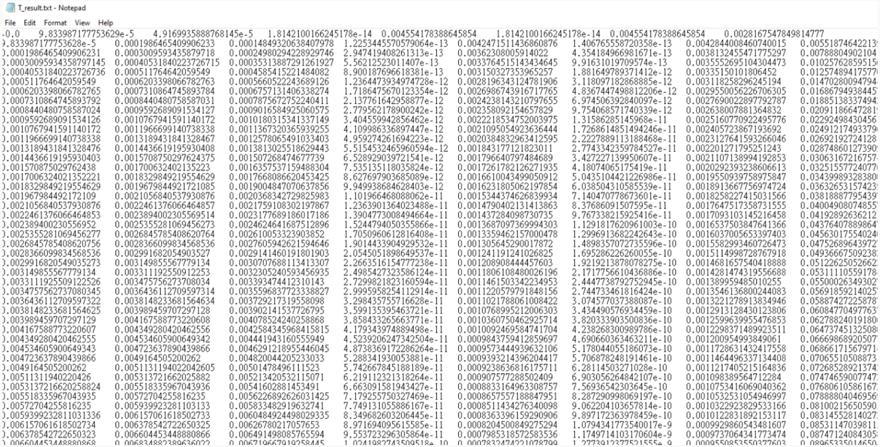
Name

results
D_beforeAdia_t=1.67.txt
D_beforeAdia_t=4.43.txt
D_beforeAdia_t=14.0.txt
D_t=0.0.txt
D_t=1.67.txt
D_t=4.43.txt
D_t=14.0.txt
Dde_beforeAdia_t=1.67.txt
Dde_beforeAdia_t=4.43.txt
Dde_beforeAdia_t=14.0.txt
GPE_beforeAdia_t=1.67.txt
GPE_beforeAdia_t=4.43.txt
GPE_beforeAdia_t=14.0.txt
GPE_t=0.0.txt
GPE_t=1.67.txt
GPE_t=4.43.txt
GPE_t=14.0.txt
M_beforeAdia_t=1.67.txt
M_beforeAdia_t=4.43.txt
M_beforeAdia_t=14.0.txt
M_t=0.0.txt
M_t=1.67.txt
M_t=4.43.txt
M_t=14.0.txt
params.txt
stepResults.txt
T_beforeAdia_t=1.67.txt
T_beforeAdia_t=4.43.txt
T_beforeAdia_t=14.0.txt
T_t=0.0.txt
T_t=1.67.txt
T_t=4.43.txt
T_t=14.0.txt

D_result.txt
GPE_result.txt
M_result.txt
T_result.txt

Output: result files

A typical result file (e.g. “T_result.txt”):



```
-0.0    9.833987177753629e-5    4.916993588768145e-5    1.8142100166245178e-14  0.004554178388645854    1.8142100166245178e-14  0.004554178388645854    0.0028167547849814777
9.833987177753628e-5    0.0001986465409906233   0.00014849320638407978  1.2253445570579064e-13  0.0042471511436860876   1.406765558720358e-13   0.0042844008460740015   0.00551874642213
).0001986465409906231   0.0003009593435879718   0.00024980294228929746  2.947419408261313e-13   0.00362308005914022     4.354184966981671e-13   0.0038132455471775297   0.00788804902104
).00030095934358797145  0.00040531840223726715  0.00035313887291261927  5.56212523011407e-13    0.0033764515143434645   9.91631019709574e-13    0.0035552695104304473   0.01025762859515
).00040531840223726736  0.000511764642059549    0.0004585415221484082   8.900187696618381e-13   0.003150327353965257    1.881649789371412e-12   0.00335150101806452     0.01257489417577
).000511764642059549    0.0006203398066782763   0.0005660522243689126   1.2364473934974728e-12  0.0028196343124781906   3.118097182868885e-12   0.003118258296245194    0.01470280094794
).0006203398066782765   0.0007310864745893784   0.0006757131406338274   1.7186475670123354e-12  0.0026986743916717765   4.8367447498812206e-12  0.0029550056226706305   0.01686794938445
).0007310864745893792   0.0008440480758587031   0.0007875672752240411   2.137761642958877e-12   0.0024238143210797655   6.974506392840097e-12   0.0027690022897792787   0.01885138337494
).0008440480758587024   0.0009592689091534127   0.0009016584925060575   2.779562178900242e-12   0.002358092154657829    9.754068571740339e-12   0.002638007881364832    0.02091186647281
).0009592689091534126   0.0010767941591140172   0.0010180315341337149   3.404559942856462e-12   0.0022218534752003975   1.31586285145968e-11    0.0025160770922495776   0.02292498430456
).0010767941591140172   0.0011966699140738338   0.0011367320365939255   4.109986336897447e-12   0.0021095054923636444   1.7268614851494246e-11  0.002405723867193692    0.02491217493379
).0011966699140738338   0.0013189431841328467   0.0012578065491033403   4.959274261694223e-12   0.0020384832963412595   2.2227889113188468e-11  0.0023127641593266046   0.02692192724128
).0013189431841328476   0.0014436619195930408   0.0013813025518629443   5.5154532465960594e-12  0.001843177121823011    2.7743342359784527e-11  0.002201271795251243    0.02874860127390
).0014436619195930403   0.0015708750297624375   0.001507268474677739    6.528929039721541e-12   0.001796640797484689    3.427227139950607e-11   0.0021107138994192853   0.03063167216757
).001570875029762438    0.001700632402135223    0.0016357537159488304   7.535135118035824e-12   0.0017261782126271935   4.18074065175419e-11    0.0020292393238606613   0.03251557724077
).0017006324021352221   0.0018329849219554629   0.0017668086620453425   8.627697903685089e-12   0.0016610043499050912   5.0435104421226986e-11  0.0019550939758975847   0.03439989328380
).0018329849219554629   0.0019679844921721085   0.0019004847070637856   9.949938684628403e-12   0.0016231805062197854   6.038504310585539e-11   0.0018913667756974724   0.03632653157423
).001967984492172109    0.0021056840537930876   0.0020368342729825983   1.101966468088062e-11   0.0015344374626839934   7.140470778673601e-11   0.0018258227415031566   0.03818887795439
).0021056840537930876   0.0022461376066464857   0.0021759108302197867   1.2363901364023488e-11  0.0014790402131413863   8.37686091507595e-11    0.0017647517358731555   0.04004908074855
).0022461376066464853   0.0023894002305569514   0.0023177689186017186   1.3904773008494664e-11  0.001437284098730735    9.767338215925416e-11   0.0017093103145216458   0.04192892636212
).002389400230556952    0.0025355281069456273   0.0024624641687512896   1.5244794050355866e-11  0.0013687097369994303   1.1291817620961003e-10  0.0016537503847641366   0.04376407889864
).0025355281069456277   0.0026845785408620764   0.002610053323903852    1.7050960612816408e-11  0.0013359462157000478   1.2996913682242643e-10  0.0016037005653397403   0.04563017554024
).0026845785408620756   0.0028366099834568536   0.0027605942621594646   1.9014433904929532e-11  0.001305645290017872    1.4898357072735596e-10  0.0015582993460726473   0.04752689643972
).0028366099834568536   0.002991682054903527    0.0029141460191801903   2.0545051898649537e-11  0.001241191241026825    1.695286226260055e-10   0.0015114998728767918   0.04936667509238
).0029916820549035273   0.003149855567779134    0.0030707688113413307   2.2663516154777238e-11  0.001208908444457603    1.9219213878078275e-10  0.0014681657540418888   0.05122625052662
).003149855567779134    0.003311192550912253    0.0032305240593456935   2.4985427323586124e-11  0.0011806108480026196   2.1717756610436886e-10  0.0014281474319556688   0.05311110559178
).0033111925509122526   0.003475756273708034    0.003393474412310143    2.7299821823160594e-11  0.0011461503342234953   2.4447738792752945e-10  0.0013899594850010255   0.05500026349302
).0034757562737080345   0.0036436112709597314   0.0035596837723338827   2.9995958254112914e-11  0.0011220579791848156   2.744733461816424e-10   0.0013546136800244083   0.05691859214025
).0036436112709597322   0.0038148233681564634   0.003729217319558098    3.298435755716628e-11   0.0011021788061008422   3.074577037388087e-10   0.0013221278913834946   0.05887427225878
).0038148233681564625   0.003989459707297128    0.003902141537726795    3.599135395463721e-11   0.0010768995212006303   3.434490576934459e-10   0.0012913128430123806   0.06084770497763
).003989459707297129    0.004167588773220608    0.004078524240258868    3.858433265663771e-11   0.0010360750462925714   3.820333903500836e-10   0.0012599639955476853   0.06278824019180
).004167588773220607    0.004349280420462556    0.0042584345968415815   4.179343974889498e-11   0.0010092469584741704   4.238268300989786e-10   0.0012298371489923511   0.06473745132508
).004349280420462555    0.004534605900649342    0.004441943160555949    4.5239206247342504e-11  0.0009843759412859697   4.690660363463211e-10   0.001200954993849061    0.06669868920507
).004534605900649343    0.004723637890439866    0.0046291218955446045   4.8738369172286264e-11  0.0009573444939632106   5.178044055186073e-10   0.0011728631432417558   0.06866171567971
).004723637890439866    0.0049164505200262      0.004820044205233033    5.288341930053881e-11   0.0009393214396204417   5.706878248191461e-10   0.0011464496337134408   0.07065510508873
).0049164505200262      0.0051131194022042605   0.00501478496111523     5.742667845188189e-11   0.0009238636816175711   6.28114503271028e-10    0.0011217405215164836   0.07268528921374
).00511311940220426    0.005313721662025882    0.005213420532115071    6.219112321318264e-11   0.000907577288502409    6.903056264842107e-10   0.001098389564712284    0.07474659007747
).0053137216620258824   0.005518335967043936    0.00541602881453491     6.663091581943427e-11   0.0008833164963308757   7.56936542303645e-10    0.0010753416069040362   0.07680610586167
).005518335967043935    0.00572704255816235     0.0056226892626031425   7.179255750327469e-11   0.0008657557188847951   8.287290998069197e-10   0.0010532531054946997   0.07888804046956
).00572704255816235     0.005939923281103133    0.005833482919632741    7.749131055886167e-11   0.0008511434276340098   9.062204103657814e-10   0.0010322923829533166   0.08100215650590
).0059399232811031336   0.006157061618502733    0.0060484924498029335   8.349682603206445e-11   0.0008363396159290906   9.897172363978459e-10   0.0010122831892153117   0.08314552814027
).006157061618502734    0.006378542722650325    0.00626780217057653     8.971694095615585e-11  0.0008204500849275294   1.0794341773540017e-9   0.0009929860543481607   0.08531147039811
).006378542722650323    0.006604453448880866    0.006491498085765594    9.553723296305864e-11   0.0007985318572583536   1.1749714103170604e-9   0.0009737064341773474   0.08747124084305
).006604453448880868    0.006834882389636022    0.006719667919258445    1.0241982743509518e-10  0.0007832474221078799   1.2773912377521555e-9   0.0009550853550377603   0.08965335014650
```

Output: result files

Many ($= numOfShells$) rows and 8 columns (from left to right):

1. Inner radius [$kpc\, h^{-1}$]
2. Outer radius [$kpc\, h^{-1}$]
3. Shell radius = (inner + outer) / 2 [$kpc\, h^{-1}$]
4. Shell mass [$10^{10}\ M_{\odot}\ h^{-1}$]
5. Shell density = shell mass / shell volume [$10^{10}\ M_{\odot}\ kpc^{-3}\ h^{2}$]
6. Enclosed mass (enclosed by outer radius) [$10^{10}\ M_{\odot}\ h^{-1}$]
7. Average density = enclosed mass / enclosed volume [$10^{10}\ M_{\odot}\ kpc^{-3}\ h^{2}$]
8. Circular velocity [$km\ s^{-1}$]

Output: use of shell radii

Different quantities may be defined using **different shell radii** (inner, outer or shell). For example, the circular velocity is defined by: $shells\_Vcir[i] = \sqrt{G\frac{shells\_enclosedMass[i]}{Tshells\_radii[i,\ 2]}}$

For consistency, plot:

- $shells\_enclosedMass[i]$ vs. $Tshells\_radii[i, 2]$ (outer radius)
- $shells\_Vcir[i]$ vs. $Tshells\_radii[i, 2]$ (outer radius)
- $shells\_mass$ vs. $Tshells\_radii[i, 3]$ (shell radius)
- etc

Adding particle types

For example, baryons such as stars ($Sshells\_mass$) and gas ($Gshells\_mass$) may be easily added to the halo sharing with the dark matter particles the same scheme of shells ($Tshells\_radii$)

Particle types can then be freely programmed to interact and evolve in various ways

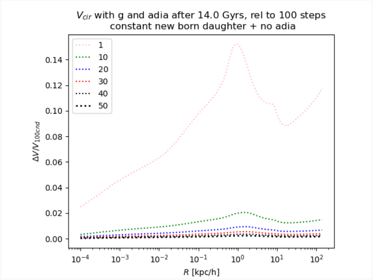
It is reminded that masses from the additional particle types must be added to the total mass profile, such as $Tshells\_enclosedMass$ for $Tshells\_GPE$ calculations

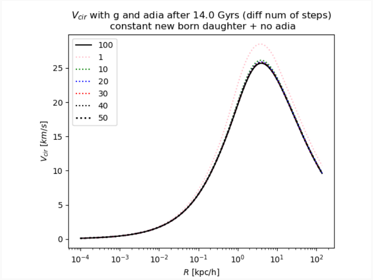
Acknowledgements

The presenters would like to thank **Prof. M.-C. Chu** for this opportunity, **Dr. Jianxiong Chen** for his guidance and **Wai-Cheong Lee**, **Kiu-Ching Leung** and **Ying Chan** for their valuable inputs.

Appendix: convergence on $numOfSteps$ (without adiabatic expansion)

Without adiabatic expansion
$(M_{vir} = 0.517,\ c = 21.6,\ \Delta_{vir} = 103.4)$:





- With kick-off effect only: $numOfSteps > 10 \Rightarrow$ good convergence

Appendix: convergence on $numOfSteps$ (with adiabatic expansion)

Convergence cannot be tested with adiabatic expansion turned on

- $numOfSteps \uparrow \Rightarrow \; \Delta t \downarrow$
- Physically, there is insufficient time for particles to reach expanded radii, especially for particles in the outer region
- In our code, **all** particles undergo adiabatic expansion **completely**

Recommended $numOfSteps = 15$