CLASS: focus on Quintessence

Miguel Zumalacárregui

Instut für Theoretische Physik - University of Heidelberg

MPA Munich - March 2014

Relevance

- Simple dynamical alternative to Λ
- Predictive: can detect/disprove with observations
- Modifications necessary for (almost) any modified gravity

Outline

1. Quintessence as a simple modification of ΛCDM

2. Implementation of Quintessence in CLASS

3. Introducing other models is easy, and will be (almost) <u>trivial</u>

The essence of Quintessence

- Scalar field ϕ
- Think as 1D particle $\phi(t)$: friction + potential

$$\frac{d^2\phi}{dt^2} + 3H\frac{d\phi}{dt} + \frac{dV}{d\phi} = 0$$

The essence of Quintessence

- Scalar field ϕ
- Think as 1D particle $\phi(t)$: friction + potential

$$\frac{d^2\phi}{dt^2} + 3H\frac{d\phi}{dt} + \frac{dV}{d\phi} = 0$$

- Can accelerate the universe

$$\begin{aligned} H &\propto \rho + \underbrace{\frac{1}{2}\left(\frac{d\phi}{dt}\right)^2 + V}_{\rho_\phi} \longrightarrow V \\ -\frac{\ddot{a}}{a} &\propto \rho + 3p + \rho_\phi + 3\underbrace{\left(\frac{1}{2}\left(\frac{d\phi}{dt}\right)^2 - V\right)}_{p_\phi} \longrightarrow -2V \end{aligned}$$

if $\phi \sim$ constant and $V(\phi) \approx \Lambda$

Quintessence Models

$$\frac{d^2\phi}{dt^2} + 3H\frac{d\phi}{dt} + \frac{dV}{d\phi} = 0$$

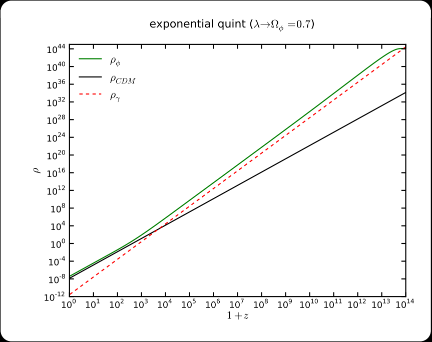
Models characerized by $V(\phi)$ + initial conditions

Exponential quintessence

- Simplest model

$$V(\phi) = M^4 \exp(-\lambda\phi/M)$$

- Attractor: $\rho_\phi \propto \rho$
- $\Rightarrow$ No acceleration



Implementation in CLASS

All modifications under suffix "_scf" = (scalar field)

- Background (in background.c)
  - parameters, indices, functions (in background.h)

```
double Omega0_scf; // scalar field energy fraction
```

Implementation in CLASS

All modifications under suffix "_scf" = (scalar field)

- ⋆ Background (in background.c)
  - parameters, indices, functions (in background.h)

```
double Omega0_scf; // scalar field energy fraction
```

  - Equations: energy & pressure density, field equation

```
background_functions(pba, a, phi, phi_prime, ...) {
 if (pba->has_scf == TRUE) {
   pvecback[pba->index_bg_phi_scf] = phi;
     ...
   pvecback[pba->index_bg_rho_scf] =
      (phi_prime*phi_prime/(2*a*a) + V_scf(pba,phi));
     ...
   rho_tot += pvecback[pba->index_bg_rho_scf];
```

Implementation in CLASS

All modifications under suffix "_scf" = (<u>sc</u>alar <u>f</u>ield)

⋆ Background (in background.c)

- parameters, indices, functions (in background.h)

```
double Omega0_scf; // scalar field energy fraction
```

- Equations: energy & pressure density, field equation

```
background_functions(pba, a, phi, phi_prime, ...) {
 if (pba->has_scf == TRUE) {
   pvecback[pba->index_bg_phi_scf] = phi;
     ...
   pvecback[pba->index_bg_rho_scf] =
      (phi_prime*phi_prime/(2*a*a) + V_scf(pba,phi));
     ...
   rho_tot += pvecback[pba->index_bg_rho_scf];
```

- <u>Initial conditions:</u> ϕ, $d\phi/dt$ relative to attractor IC
- <u>Potential and its derivatives</u>: V, V', V''

Implementation in CLASS II

Getting the right Ω_ϕ

No simple relation $\phi(a) \rightarrow$ need to solve equations

- choose a starting parameter α_{try}, obtain $\Omega_{\phi,\text{try}}$
- vary α_{try} it until $|\Omega_{\phi,\text{try}} - \Omega_{\phi,\text{wish}}| <$desired precission

Requires undersanding of model & parameters

Implementation in CLASS II

Getting the right Ω_ϕ

No simple relation $\phi(a) \rightarrow$ need to solve equations

- choose a starting parameter α_{try}, obtain $\Omega_{\phi,\text{try}}$
- vary α_{try} it until $|\Omega_{\phi,\text{try}} - \Omega_{\phi,\text{wish}}|$ <desired precission

Requires undersanding of model & parameters

$\star$ in class.c (for exponential quintessence)

```
#define TUNE_PARAM pba->scf_lambda
int tune_scalar_field_parameters(* ppr, * pba ) {
  ...
  while (fabs(pba->Omega0_scf-Omega0_scf_try) > tolerance){

    TUNE_PARAM=0.5*(scf_TUNE_PARAM_max+scf_TUNE_PARAM_min);

    if (background_init(ppr,pba) == FAILURE) { ... }

    Omega0_scf_try = (value from struct background)
    (change scf_TUNE_PARAM_max and scf_TUNE_PARAM_min)
```

Implementation in CLASS III

- ⋆ Perturbations (in perturbations.c)
  - indices for perturbations (in perturbations.h)
  - Equations: energy & pressure density, field equation

```
if (pba->has_scf == TRUE) {
  ppw->delta_rho += ppw->pvecback[pba->index_bg_dV_scf
      ]*y[ppw->pv->index_pt_phi_scf] + ...;
```

  - Initial conditions: $\delta\phi$, $\delta d\phi/dt$ set to zero

```
if (pba->has_scf == TRUE) {
  ppw->pv->y[ppw->pv->index_pt_phi_scf] = 0.;
```

- ⋆ Other modifications
  - read parameters (in input.c)

```
class_call(parser_read_double(pfc,"scf_lambda",...)...)
```

  - output (in output.c)

Usage and output

⋆ Input file: choose $\Omega_{\rm scf}$, parameters, initial conditions...

- Λ + early DE: ⟶

```
Omega0_scf = .05
```

- only scalar field:

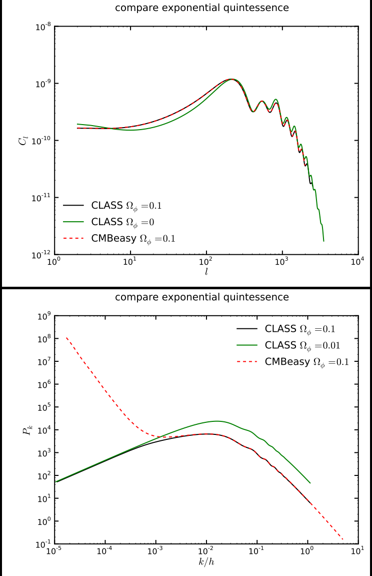
```
Omega_Lambda = 0
Omega_fld = 0
```

- large initial ϕ, ϕ':

```
initial_phi_scf = 100
initial_phi_prime_scf = 100
```

- parameters:

```
scf_lambda = 8
scf_A = 0.01
scf_B = 34.8
scf_alpha = 2
```

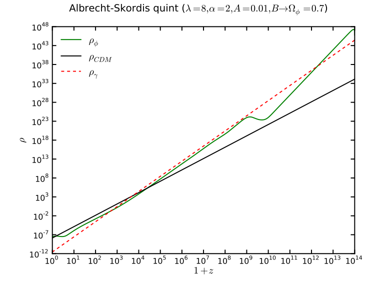


⋆ Plot the output!

Introducing a new model

Exponential pot. with a bump:

$$
\begin{aligned}
V(\phi) &= M^4\left((\phi - B)^\alpha + A\right) e^{-\lambda\phi/M} \\
&= V_p(\phi) V_e(\phi)
\end{aligned}
$$



Modify:

- potential + derivatives
- tuning function
- (initial conditions)

```
double V_scf(*pba, phi) //Potential V = polynomial*exp
  return  V_p_scf(pba,phi)*V_e_scf(pba,phi);

double V_p_scf(*pba, phi) // polynomial factor of V
  return  pow(phi-pba->scf_B,pba->scf_alpha) + pba->scf_A;
  ...
#define TUNE_PARAM pba->scf_B
int tune_scalar_field_parameters(* ppr, * pba ) {
  ...
```

The Future...

$\exists$ Many theories: Quintessence, K-essence, Scalar-Tensor/$f(R)$, Galileons... $\boxed{\in \text{Horndeski theory}}$ $\quad X = -(\partial\phi)^2/2$

$$\mathcal{L}_H = G_2(X,\phi) - G_3(X,\phi)\Box\phi + G_4 R + G_{4,X}\left[(\Box\phi)^2 - \phi_{;\mu\nu}\phi^{;\mu\nu}\right]$$

$$+G_5 G_{\mu\nu}\phi^{;\mu\nu} - \frac{G_{5,X}}{6}\left[(\Box\phi)^3 - 3(\Box\phi)\phi_{;\mu\nu}\phi^{;\mu\nu} + 2\phi_{;\mu}{}^{;\nu}\phi_{;\nu}{}^{;\lambda}\phi_{;\lambda}{}^{;\mu}\right]$$

Similar modifications, but very complicated equations!

The Future...

$\exists$ Many theories: Quintessence, K-essence, Scalar-Tensor/$f(R)$, Galileons... $\boxed{\in \text{Horndeski theory}}$ $X = -(\partial\phi)^2/2$

$$\mathcal{L}_H = G_2(X,\phi) - G_3(X,\phi)\Box\phi + G_4 R + G_{4,X}\left[(\Box\phi)^2 - \phi_{;\mu\nu}\phi^{;\mu\nu}\right]$$

$$+G_5 G_{\mu\nu}\phi^{;\mu\nu} - \frac{G_{5,X}}{6}\left[(\Box\phi)^3 - 3(\Box\phi)\phi_{;\mu\nu}\phi^{;\mu\nu} + 2\phi_{;\mu}{}^{;\nu}\phi_{;\nu}{}^{;\lambda}\phi_{;\lambda}{}^{;\mu}\right]$$

Similar modifications, but very complicated equations!

Hi-CLASS

- Horndeski in CLASS
- Equations alerady worked out - Bellini & Sawicki arxiv:1403.xxxx
- User only needs to provide $\{G_i(X,\phi)\}$ (+ tuning + IC): all the terms in the eqs. computed and implemented automatically by a Python wrapper

Conclusions

- Quintessence is a simple, dynamical alternative to Λ (analogous to inflationary models)
- It provides all the basic modifications needed to include *almost* any theory of gravity in CLASS
  - All modifications can be searched through the suffix "_scf"
  - To be included in a next version of CLASS, can be downloaded from https://github.com/miguelzuma/class_scf
- Easy to include any other quintessence model. Just change:
  - Potential & derivatives
  - Initial conditions
  - Tuning algorithm
- Hi-CLASS: soon any* scalar-tensor gravity in CLASS

* Except higher dimensional (e.g. DGP), non-local, Lorentz violating or non 2nd-order Euler-Lagrange variations (e.g. arXiv:1308.4685)