Two-Step Bayesian PINNs for Uncertainty Estimation

Pablo Flores  
Pontificia Universidad Católica de Chile  
ptflores1@uc.cl

Olga Graf  
Technical University of Munich  
graf@ma.tum.de

Pavlos Protopapas  
Harvard University  
pavlos@seas.harvard.edu

Karim Pichara  
Pontificia Universidad Católica de Chile  
kpb@ing.puc.cl

Abstract

We use a two-step procedure to train Bayesian neural networks that provide uncertainties over the solutions to differential equation systems provided by Physics-Informed Neural Networks (PINNs). We take advantage of available error bounds over PINNs to formulate a heteroscedastic variance that improves the uncertainty estimation. Furthermore, we solve forward problems and utilize the uncertainties obtained to improve parameter estimation in inverse problems in the fields of cosmology and fermentation.

1 Introduction

Physics-Informed Neural Networks (PINNs) are a type of neural network (NN) first proposed by [25] that are trained to solve a differential equation (DE) by incorporating the underlying physics of the problem into the network architecture. This allows the network to learn the DE solution without needing additional data. PINNs present many advantages over traditional numerical solvers; they are continuous, differentiable, and parallelizable, i.e., they do not need to compute previous time steps in order to compute the next one. Although these advantages are noteworthy, the task of computing solution errors or error bounds remains largely unresolved, with ongoing research in this area [28][29][15].

Since the first appearance of PINNs, there has been a rapid growth of publications that cover a wide range of applications. Heat transfer [6], wave equations [39] and fluid mechanics [23][8][30] are some examples. There has also been interest in identifying and overcoming failure modes and optimization challenges [38][42][24][14]. Bayesian PINNs have also been explored [45][27][20].

Given the lack of advances for estimating errors, we leverage Bayesian NNs to obtain uncertainties over the solutions. The contributions of our work are the following:

- We propose an error bound based heteroscedastic variance that improves uncertainty quality.
- We solve forward problems for equations in the fields of cosmology and fermentation processes.
- Finally, we utilize solution bundles [17] to solve inverse problems, i.e., do parameter estimation over the equation parameters.

2 Background

2.1 Problem Formulation

We adopt a slightly different formulation from [38]. The DEs we will work with can be defined as follows:

\[\mathcal{F}_\lambda[u(x)] = f(x), \quad x \in \Omega,\]
\[\mathcal{B}_\lambda[u(x)] = b(x), \quad x \in \Gamma,\]

where \(x\) is the space-time coordinate, \(\Omega\) is a bounded domain with boundary \(\Gamma\), \(f(x)\) is the source term (also called forcing function), \(u\) is the solution of the system, \(\mathcal{F}_\lambda\) is a differential operator, \(\mathcal{B}_\lambda\) and \(b(x)\) are the initial condition (IC) or boundary condition (BC) operator and term, respectively, and \(\lambda\) denotes the parameters of the system.

In this paper, we will focus on problems where the operators \(\mathcal{F}_\lambda, \mathcal{B}_\lambda\) and terms \(f(x), b(x)\) are known. If \(\lambda\) is assumed to be known, the goal is to find the solution \(u\), which is referred to as the forward problem. Conversely, if \(u\) is known and the aim is to estimate \(\lambda\), then this is known as the inverse problem.

2.2 Physics-Informed Neural Networks

A Physics-Informed Neural Network uses a deep neural network, \(u_\theta(x)\), to approximate the true solution \(u(x)\). The network is trained to minimize the residual \(r_\theta(x) = \mathcal{F}_\lambda[u_\theta(x)] - f(x)\), aiming for \(r_\theta(x) = 0\).

The mean squared error is commonly used as the loss function. A transformation \(\tilde{u}_\theta(x)\) ensures that initial conditions are met (we use the transformation shown in [20, 9]). The optimization problem is:

\[
\min_\theta \sum_i^N r^2_\theta(x_i).
\]

Solution bundles [17] extend PINNs by allowing the network to take system parameters \(\lambda \in \Lambda\) as inputs. This eliminates the need for multiple trainings for different \(\lambda\) values, and the optimization objective becomes: \(\min_\theta \sum_i^N \sum_j^M r^2_\theta(x_i, \lambda_j)\).

2.3 Uncertainty Quantification with Bayesian Neural Networks

To quantify uncertainty in NNs, we will employ the Bayesian approach. There are a number of different methods for posterior \(p(\theta|D)\) calculation. In this paper, we will focus on the following three:

Bayes By Backpropagation (BBB) [3] employs variational inference to approximate the posterior. It assumes a variational distribution \(q(\theta|\rho)\) and minimizes its Kullback-Leibler divergence from the true posterior.

Hamiltonian Monte Carlo (HMC) [16] draws samples from the posterior distribution using a Markov chain. Specifically, it is a Metropolis-Hastings algorithm [31, 21] that generates proposals by simulating the movement of a particle using Hamiltonian dynamics [37]. We will use the No-U-Turn Sampler (NUTS) [22] which is an extension of HMC.

Neural Linear Model (NLM) [41] is a Bayesian linear regression model with learned feature basis. Given specific priors and likelihood choices, analytical forms for both the posterior \(p(\theta|D)\) and the posterior predictive \(p(y|x)\) can be derived.

2.4 Error Bounds for PINNs

Good quality uncertainties should correlate with the true error of a solution. Since true error is not accessible, in Section 3.2 we use error bounds to improve the uncertainty in the Bayesian NNs.

In [28][29], the authors present algorithms for calculating error bounds on PINNs. These bounds are applicable to linear ODEs, systems of linear ODEs, non-linear ODEs in the form \(\epsilon v^1\), as well as certain types of PDEs. These algorithms are independent of the NN architecture and depend solely on the structure of the equation as defined in Eq. (1) and the residuals of the DE.

\[\epsilon \text{ is a variable and } |\epsilon| \ll 1.\]
The network error is denoted as $\eta(x) := u(x) - \tilde{u}_\theta(x)$, and the error bound is represented by a scalar function $B$ such that $||\eta(x)|| \leq B(x)$.

3 Two-Step Bayesian PINNs

We use a two-step approach to obtain uncertainties in the solutions of equations. In the first step, we train a PINN as a bundle network (Section 2.2) that we refer to as the deterministic network. We denote this network as $u_{\theta_{det}}: \Omega, \Lambda \rightarrow \mathbb{R}$.

3.1 Bayesian Neural Network Training

The second step involves training a Bayesian neural network using the outputs of $u_{\theta_{det}}$ as targets.

Dataset: We define a dataset by taking the space-time coordinates as independent variables, the equation parameters as additional input variables, and the outputs of the deterministic net as dependent variables $D = \{(x_i, \lambda_\theta, u_{\theta_{det}}(x_i, \lambda)) \mid x_i \in \Omega, \lambda_\theta \in \Lambda, u_{\theta_{det}}(x_i, \lambda) \in \mathbb{R}\}_{i=1}^{N'}$.

Distributions: We define a Bayesian neural network $u_\theta$ by assigning distributions to the network parameters and dataset as it is done in [20]. For all three methods described in Section 2.3 we use the following distributions:

$$\theta \sim \mathcal{N}(0, \sigma_{prior}),$$

$$u_{\theta_{det}}(\hat{x}, \hat{\lambda}) \mid \hat{x}, \hat{\lambda}, \theta \sim \mathcal{N}(u_{\theta}(\hat{x}, \hat{\lambda}), \sigma_{\text{Likelihood}}(\hat{x}, \hat{\lambda})).$$

Given Eq. (4), we define the likelihood as $p(D \mid \theta) = \prod_{i=1}^{N'} p(u_{\theta_{det}}(x_i, \lambda_i) \mid x_i, \lambda_i, \theta)$.

For BBB, we also define the variational posterior distribution using the mean field approximation [2]. For NLM, we can obtain the posterior and posterior predictive distributions through analytical derivation:

$$u \mid x, \lambda, D \sim \mathcal{N}(\Phi_\theta(x, \lambda)\mu_{post}, \sigma_{\text{Likelihood}}^2(x, \lambda) + \Phi_\theta(x, \lambda)\Sigma_{post}\Phi_\theta^T(x, \lambda)),$$

where $\mu_{post}$ and $\Sigma_{post}$ are the posterior parameters and $\Phi_\theta(x, \lambda)$ is the learned feature map, we provide details in Section VII.I.I of the supplementary material.

3.2 Heteroscedastic Variance Based on Error Bounds

Bayesian NNs are usually trained using homoscedastic variance, i.e., $\sigma_{\text{Likelihood}}(x, \lambda) = \text{const}$. We take advantage of the error bounds described in Section 2.4 to define a heteroscedastic variance $\sigma_{\text{Likelihood}}(x, \lambda) = B(x, \lambda)$.

Since the error bounds are not available for all equations, we use a slightly modified homoscedastic variance in the remaining equations; it is defined as:

$$\sigma_{\text{Likelihood}}(x, \lambda) = \begin{cases} 0 & x \in \Gamma, \\ \text{const} & x \notin \Gamma, \end{cases}$$

which means that we set to zero the variance at the coordinates of ICs. The reason for this is the enforcement of ICs, thus we are certain that the solution is correct at those coordinates.

3.3 Uncertainty Computation

The calculation of uncertainty is straightforward as it is the standard deviation of the posterior predictive distribution. We can obtain it by applying the law of total variance, its derivation can be found in Section VII.I.II of the supplementary material.

4 Parameter Estimation for Inverse Problems

We can use Bayesian framework to estimate the parameters $\lambda$ of a DE system. For a given set of observations $O = \{(x_i, \mu_i, \sigma_i) \mid x_i \in \Omega, \mu_i \in \mathbb{R}, \sigma_i \in \mathbb{R^+}\}_{i=1}^{O}$, we seek to find the posterior
distribution \( p(\lambda|O) \) of the parameters. Here \( \mu_i \) and \( \sigma_i \) are the mean and standard deviation of the true solution, respectively.

In this work, we assume a uniform prior over \( \lambda \). To define the likelihood \( p(O|\lambda) \), we assume the observations have a normal distribution around the true solution, i.e., \( \mu_i \sim \mathcal{N}(a(x_i), \sigma_i) \). Since in Section 3 we obtain a distribution over all possible solutions, we need to marginalize over them to get the likelihood, the details of this marginalization are provided in Section VII.I.III of the supplementary material. Having defined the prior and likelihood, we can apply a sampling algorithm to approximate \( p(\lambda|O) \). We use the emcee Python library [18] which implements the samplers introduced in [19].

5 Experiments and discussion

In this section we present the results we obtained for solving the forward problem using the standard PINN as described in Section 3 and inverse problem using bundle solutions and estimating the parameters as described in Sections 2.2 and 4, respectively.

**Cosmology** We address the same four cosmological models as in [8]. Details of the equations are given in Section VII.I.I of the Supplementary Material. In Fig. 1 we present the solutions we obtained using the three Bayesian methods described in Section 2.3. To solve this forward problem we select fixed values for the equation parameters.

We estimated the parameters of the equation for the \( \Lambda CD\) model as explained in Section 4. The \( \Lambda CD\) model has only one parameter for which we obtained 10,000 samples for each Bayesian NN and also for a deterministic net. The means and standard deviations obtained were: 0.25 ± 0.025, 0.22 ± 0.0024, 0.20 ± 0.012, 0.1 ± 0.0001 for the deterministic net, NLM, BBB, and HMC, respectively.

**Fermentation:** We solve equations for Lactic Acid Bacteria (LAB) fermentation and cacao fermentation proposed in [44] [32], respectively. Details of the equations are given in Section VII.I.II of the Supplementary Material. In Fig. 2 we present the solutions to the forward problem for the LAB system.

6 Conclusions

We successfully applied a two-step procedure to train a Bayesian NN and proposed a heteroscedastic variance based on error bounds. We have shown that the use of this heteroscedastic variance effectively improves uncertainty quality on equations from cosmological and fermentation dynamical models. This emphasizes the need for methods that provide PINNs’ error bounds for any DE class.
The results show that when using homoscedastic variance BBB heavily underestimates the uncertainty while NLM and HMC tend to increase their uncertainty as the solution moves away from the training region. We then utilized the solutions to estimate the parameter of the ΛCDM model and observed that Bayesian solutions yield a less disperse estimation than deterministic nets.

In future work we intend to do parameter estimation for the remaining DEs and analyze in depth the benefits of using well calibrated uncertainties.

References


VII  Supplementary Material

VII.I  Bayesian Methods Details

VII.I.I  NLM Posterior Predictive Parameters

\[
\Sigma_{\text{post}} = \left( \Phi^T \Sigma \Phi(x, \lambda) + \sigma^2_{\text{prior}} I \right)
\]
\[
\mu_{\text{post}} = \Sigma_{\text{post}} \left( \Sigma^{-1} \Phi \theta(x, \lambda) \right)^T u_{\theta_{\text{det}}}(x, \lambda, D)
\]

(7)

(8)

VII.I.II  Law of Total Variance Derivation

\[
\text{Var}(u|x, \lambda, \mathcal{D}) = \mathbb{E}_{\theta|\mathcal{D}} \left[ \text{Var}(u|x, \lambda, \theta) \right] + \text{Var}_{\theta|\mathcal{D}} \left[ \mathbb{E}(u|x, \lambda, \theta) \right]
\]
\[
= \mathbb{E}_{\theta|\mathcal{D}} \left[ \sigma^2_{\text{likelihood}}(x, \lambda) \right] + \text{Var}_{\theta|\mathcal{D}} \left[ u_{\theta}(x, \lambda) \right]
\]
\[
\approx \sigma^2_{\text{likelihood}}(x, \lambda) + \frac{1}{M} \sum_{i=1}^{M} (u_{\theta_i}(x, \lambda) - \bar{u}(x, \lambda))^2,
\]

(9)

where \( \theta_i \sim p(\theta|\mathcal{D}) \) and \( \bar{u}(x, \lambda) \approx \frac{1}{M} \sum_{i=1}^{M} u_{\theta_i}(x, \lambda) \) is the sample mean of the net outputs.

The latter approximation applies to BBB and NUTS, for BBB the samples are taken from the variational posterior and for NUTS we use the samples of the true posterior the method yields. For NLM we have the analytical expression in Eq. (7).

VII.I.III  Marginalization Over Solutions

\[
p(x_i, \mu_i, \sigma_i|\lambda) = \int_{\mathcal{U}} \mathcal{N}(\mu_i; u(x_i, \lambda), \sigma_i) \cdot p(u(x_i, \lambda)|x_i, \lambda, \mathcal{D}) du
\]
\[
\approx \frac{1}{M} \sum_{j=1}^{M} \mathcal{N}(\mu_i; u^{(j)}(x_i, \lambda), \sigma_i), \text{ where } u^{(j)} \sim p(u(x_i, \lambda)|x_i, \lambda, \mathcal{D}),
\]

(10)

\[
p(O|\lambda) = \prod_{i=1}^{O} p(x_i, \mu_i, \sigma_i|\lambda).
\]

(11)

VII.II  Equation details

We now describe the equations, initial conditions and parameters of each dynamical model.

VII.I.I  Cosmology

In this section we specify the cosmological equations we used. These equations were solved with PINNs in [8].

\( \Lambda \)CDM

Equation:

\[
\frac{dx_m}{dz} = \frac{3x_m}{1 + z}
\]

(12)

Initial conditions:

\[
x_m(z) = \Omega_{m,0}
\]

(13)

Parameters:

\[
\Omega_{m,0}
\]

(14)

Hubble function:

\[
H(z) = H_0 \sqrt{x_m(z) + 1 - \Omega_{m,0}}
\]

(15)
Parametric dark energy

Equation:
\[
\frac{dx_{DE}}{dz} = 3x_{DE} \left( 1 + w_0 + \frac{w_1 z}{1+z} \right)
\]  
(16)

Initial conditions:
\[x_m(z) = 1 - \Omega_{m,0}\]  
(17)

Parameters:
\[w_0, w_1, \Omega_{m,0}\]  
(18)

Hubble function:
\[H(z) = H_0 \sqrt{\Omega_{m,0}(1+z)^3 + x_{DE}(z)}\]  
(19)

Quintessence

Equation:
\[
\frac{dx}{dN} = -3x + \sqrt{6} \frac{\lambda y^2}{2} + \frac{3}{2}x(1+x^2-y^2)
\]  
(20)

\[
\frac{dy}{dN} = \sqrt{6} \frac{xy\lambda}{2} + \frac{3}{2}y(1+x^2-y^2)
\]  
(21)

Initial conditions:
\[x_0 = 0\]  
\[y_0 = \frac{1 - \Omega^\Lambda_{m,0}}{\Omega^\Lambda_{m,0}(1+z_0)^3 + 1 - \Omega^\Lambda_{m,0}}\]  
(22)

Parameters:
\[H_0, \Omega^\Lambda_{m,0}\]  
(24)

Hubble function:
\[H(z) = H_0 \sqrt{\frac{\Omega^\Lambda_{m,0}(1+z)^3}{1-x^2-y^2}}\]  
(25)

\(f(R)\) gravity

Equation:
\[
\frac{dx}{dz} = \frac{1}{z+1}(-\Omega - 2v + x + 4y + xv + x^2)
\]  
(26)

\[
\frac{dy}{dz} = \frac{-1}{z+1}(xz\Gamma - xy + 4y - 2yv)
\]  
(27)

\[
\frac{dv}{dz} = \frac{-v}{z+1}(x\Gamma + 4 - 2v)
\]  
(28)

\[
\frac{d\Omega}{dz} = \frac{\Omega}{z+1}(-1 + 2v + x)
\]  
(29)

\[
\frac{dr}{dz} = \frac{r\Gamma x}{z+1}
\]  
(30)

\[
\text{where } \Gamma(r) = \frac{(r + b) \left[ (r + b)^2 - 2b \right]}{4br}
\]  
(31)

Initial conditions:
\[x_0 = 0\]  
(32)

\[
y_0 = \frac{\Omega_{m,0}(1+z_0)^3 + 2(1 - \Omega^\Lambda_{m,0})}{2 [\Omega_{m,0}(1+z_0)^3 + (1 - \Omega^\Lambda_{m,0})]}
\]  
(33)

\[
v_0 = \frac{\Omega_{m,0}(1+z_0)^3 + 4(1 - \Omega^\Lambda_{m,0})}{2 [\Omega_{m,0}(1+z_0)^3 + (1 - \Omega^\Lambda_{m,0})]}
\]  
(34)

\[
\Omega_0 = \frac{\Omega_{m,0}(1+z_0)^3}{\Omega_{m,0}(1+z_0)^3 + (1 - \Omega^\Lambda_{m,0})}
\]  
(35)

\[
r_0 = \frac{\Omega_{m,0}(1+z_0)^3 + 4(1 - \Omega^\Lambda_{m,0})}{1 - \Omega^\Lambda_{m,0}}
\]  
(36)

Parameters:
\[b, \Omega^\Lambda_{m,0}, H_0^\Lambda\]  
(37)

Hubble function:
\[H(z) = H_0^\Lambda \sqrt{\frac{r}{2v(1 - \Omega^\Lambda_{m,0})}}\]  
(38)
VII.II.2 Fermentation

Lactic acid bacteria This system was proposed in [44].

Equations:
\[
\frac{dx}{dt} = \alpha x - \beta x^2 - \gamma xy^2 - \tau xz^2 \tag{39}
\]
\[
\frac{dy}{dt} = \delta y - \theta y^2 - \rho yz \tag{40}
\]
\[
\frac{dz}{dt} = \sigma z - \phi z^2 - \omega xz \tag{41}
\]

Initial conditions:
\[(x, y, z) = (7.5797, 6.44, 1.9) \tag{42}\]

Parameters:
\[\alpha, \beta, \gamma, \tau, \theta, \rho, \sigma, \phi, \omega \tag{43}\]

Cocoa This system was proposed in [32]. We present the meaning and initial conditions of each variable in Table 1 and the specification of the system’s parameters in Table 2.

Equations:
\[
\frac{dx}{dt} = -\gamma_{\text{Glc}} x \frac{\mu_{\text{Glc}}}{x + K_{\text{Glc}}} w - \gamma_{\text{LAB}} \frac{\mu_{\text{LAB}}}{x + K_{\text{LAB}}} r \tag{45}
\]
\[
\frac{dy}{dt} = -\gamma_{\text{Fru}} y \frac{\mu_{\text{Fru}}}{y + K_{\text{Fru}}} w \tag{46}
\]
\[
\frac{dz}{dt} = \gamma_{\text{EtOH}} y \frac{\mu_{\text{EtOH}}}{x + K_{\text{Glc}}} w + \gamma_{\text{EtOH}} x \frac{\mu_{\text{EtOH}}}{y + K_{\text{Fru}}} w - \gamma_{\text{LAB}} \frac{\mu_{\text{LAB}}}{z + K_{\text{LAB}}} s \tag{47}
\]
\[
\frac{du}{dt} = \gamma_{\text{LA}} \frac{\mu_{\text{LAB}}}{x + K_{\text{Glc}}} r - \gamma_{\text{LAB}} \frac{\mu_{\text{LAB}}}{u + K_{\text{LAB}}} s \tag{48}
\]
\[
\frac{dv}{dt} = \gamma_{\text{Ac}} \frac{\mu_{\text{LAB}}}{x + K_{\text{Glc}}} r + \gamma_{\text{Ac}} \frac{\mu_{\text{LAB}}}{x + K_{\text{LAB}}} r \tag{49}
\]
\[
\frac{dw}{dt} = \frac{\mu_{\text{Glc}}}{x + K_{\text{Glc}}} w + \frac{\mu_{\text{Fru}}}{y + K_{\text{Fru}}} w - k_y w z \tag{50}
\]
\[
\frac{dr}{dt} = \frac{\mu_{\text{LAB}}}{x + K_{\text{Glc}}} r - k_{LAB} r u \tag{51}
\]
\[
\frac{ds}{dt} = \frac{\mu_{\text{AAB}}}{z + K_{\text{AB}}} s + \frac{\mu_{\text{AAB}}}{u + K_{\text{LAB}}} s - k_{AAB} s^2 \tag{52}
\]

Table 1: Cocoa fermentation process variables meaning and initial conditions

<table>
<thead>
<tr>
<th>Variable</th>
<th>Meaning</th>
<th>Initial Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>[Glc]</td>
<td>51.963</td>
</tr>
<tr>
<td>y</td>
<td>[Fru]</td>
<td>57.741</td>
</tr>
<tr>
<td>z</td>
<td>[EtOH]</td>
<td>0</td>
</tr>
<tr>
<td>u</td>
<td>[LA]</td>
<td>0</td>
</tr>
<tr>
<td>v</td>
<td>[Ac]</td>
<td>0</td>
</tr>
<tr>
<td>w</td>
<td>[Y]</td>
<td>0.029180401</td>
</tr>
<tr>
<td>r</td>
<td>[LAB]</td>
<td>0.007868827</td>
</tr>
<tr>
<td>s</td>
<td>[AAB]</td>
<td>3.36634e-6</td>
</tr>
</tbody>
</table>
Table 2: Cocoa fermentation process parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu_{\text{Y, Glc}}$</td>
<td>h^{-1}</td>
<td>Maximum specific growth rate of Y on Glc</td>
</tr>
<tr>
<td>$\mu_{\text{Y, Fru}}$</td>
<td>h^{-1}</td>
<td>Maximum specific growth rate of Y on Fru</td>
</tr>
<tr>
<td>$\mu_{\text{LAB}}$</td>
<td>h^{-1}</td>
<td>Maximum specific growth rate of LAB on Glc</td>
</tr>
<tr>
<td>$\mu_{\text{AAB, EtOH}}$</td>
<td>h^{-1}</td>
<td>Maximum specific growth rate of AAB on EtOH</td>
</tr>
<tr>
<td>$\mu_{\text{AAB, LA}}$</td>
<td>h^{-1}</td>
<td>Maximum specific growth rate of AAB on LA</td>
</tr>
<tr>
<td>$K_{\text{Y, Glc}}$</td>
<td>mg(Glc) g(pulp)^{-1}</td>
<td>Substrate saturation constant of Y growth on Glc</td>
</tr>
<tr>
<td>$K_{\text{Y, Fru}}$</td>
<td>mg(Fru) g(pulp)^{-1}</td>
<td>Substrate saturation constant of Y growth on Fru</td>
</tr>
<tr>
<td>$K_{\text{LAB, Glc}}$</td>
<td>mg(Glc) g(pulp)^{-1}</td>
<td>Substrate saturation constant of LAB growth on Glc</td>
</tr>
<tr>
<td>$K_{\text{AAB, EtOH}}$</td>
<td>mg(EtOH) g(pulp)^{-1}</td>
<td>Substrate saturation constant of AAB growth on EtOH</td>
</tr>
<tr>
<td>$K_{\text{AAB, LA}}$</td>
<td>mg(LA) g(pulp)^{-1}</td>
<td>Substrate saturation constant of AAB growth on LA</td>
</tr>
<tr>
<td>$k_{\text{Y}}$</td>
<td>mg(EtOH)^{-1} h^{-1}</td>
<td>Mortality rate constant of Y</td>
</tr>
<tr>
<td>$k_{\text{LAB}}$</td>
<td>mg(LA)^{-1} h^{-1}</td>
<td>Mortality rate constant of LAB</td>
</tr>
<tr>
<td>$k_{\text{AAB}}$</td>
<td>mg(Ac)^{-2} h^{-1}</td>
<td>Mortality rate constant of AAB</td>
</tr>
<tr>
<td>$Y_{\text{Glc</td>
<td>Y}}$</td>
<td>mg(Glc) mg(Y)^{-1}</td>
</tr>
<tr>
<td>$Y_{\text{Glc</td>
<td>LAB}}$</td>
<td>mg(Glc) mg(LAB)^{-1}</td>
</tr>
<tr>
<td>$Y_{\text{Fru</td>
<td>Y}}$</td>
<td>mg(Fru) mg(Y)^{-1}</td>
</tr>
<tr>
<td>$Y_{\text{EtOH</td>
<td>Glc</td>
<td>Y}}$</td>
</tr>
<tr>
<td>$Y_{\text{EtOH</td>
<td>Fru</td>
<td>Y}}$</td>
</tr>
<tr>
<td>$Y_{\text{EtOH</td>
<td>AAB}}$</td>
<td>mg(EtOH) mg(AAB)^{-1}</td>
</tr>
<tr>
<td>$Y_{\text{LA</td>
<td>LAB}}$</td>
<td>mg(LA) mg(LAB)^{-1}</td>
</tr>
<tr>
<td>$Y_{\text{LA</td>
<td>AAB}}$</td>
<td>mg(LA) mg(AAB)^{-1}</td>
</tr>
<tr>
<td>$Y_{\text{Ac</td>
<td>LAB}}$</td>
<td>mg(Ac) mg(LAB)^{-1}</td>
</tr>
<tr>
<td>$Y_{\text{Ac</td>
<td>AAB}}$</td>
<td>mg(Ac) mg(AAB)^{-1}</td>
</tr>
<tr>
<td>$Y_{\text{LA</td>
<td>Ac</td>
<td>AAB}}$</td>
</tr>
</tbody>
</table>
Table 3: Measurements of the Hubble parameter $H$ using the Cosmic Chronometers technique

<table>
<thead>
<tr>
<th>$z$</th>
<th>$H(z) \pm \sigma_H$ $[\text{km/s Mpc}]$</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.09</td>
<td>69 ± 12</td>
<td></td>
</tr>
<tr>
<td>0.17</td>
<td>83 ± 8</td>
<td></td>
</tr>
<tr>
<td>0.27</td>
<td>77 ± 14</td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>95 ± 17</td>
<td></td>
</tr>
<tr>
<td>0.9</td>
<td>117 ± 23</td>
<td>[40]</td>
</tr>
<tr>
<td>1.3</td>
<td>168 ± 17</td>
<td></td>
</tr>
<tr>
<td>1.43</td>
<td>177 ± 18</td>
<td></td>
</tr>
<tr>
<td>1.53</td>
<td>140 ± 14</td>
<td></td>
</tr>
<tr>
<td>1.75</td>
<td>202 ± 40</td>
<td></td>
</tr>
<tr>
<td>0.48</td>
<td>97 ± 62</td>
<td>[43]</td>
</tr>
<tr>
<td>0.88</td>
<td>90 ± 40</td>
<td></td>
</tr>
<tr>
<td>0.1791</td>
<td>75 ± 4</td>
<td></td>
</tr>
<tr>
<td>0.1993</td>
<td>75 ± 5</td>
<td></td>
</tr>
<tr>
<td>0.3519</td>
<td>83 ± 14</td>
<td></td>
</tr>
<tr>
<td>0.5929</td>
<td>104 ± 13</td>
<td>[34]</td>
</tr>
<tr>
<td>0.6797</td>
<td>92 ± 8</td>
<td></td>
</tr>
<tr>
<td>0.7812</td>
<td>105 ± 12</td>
<td></td>
</tr>
<tr>
<td>0.8754</td>
<td>125 ± 17</td>
<td></td>
</tr>
<tr>
<td>1.037</td>
<td>154 ± 20</td>
<td></td>
</tr>
<tr>
<td>0.07</td>
<td>69 ± 19.6</td>
<td></td>
</tr>
<tr>
<td>0.12</td>
<td>68.6 ± 26.2</td>
<td>[12]</td>
</tr>
<tr>
<td>0.2</td>
<td>72.9 ± 29.6</td>
<td></td>
</tr>
<tr>
<td>0.28</td>
<td>88.8 ± 36.6</td>
<td></td>
</tr>
<tr>
<td>1.363</td>
<td>160 ± 33.6</td>
<td>[33]</td>
</tr>
<tr>
<td>1.965</td>
<td>186.5 ± 50.4</td>
<td></td>
</tr>
<tr>
<td>0.3802</td>
<td>83 ± 13.5</td>
<td></td>
</tr>
<tr>
<td>0.4004</td>
<td>77 ± 10.2</td>
<td></td>
</tr>
<tr>
<td>0.4247</td>
<td>87.1 ± 11.2</td>
<td>[35]</td>
</tr>
<tr>
<td>0.4497</td>
<td>92.8 ± 12.9</td>
<td></td>
</tr>
<tr>
<td>0.4783</td>
<td>80.9 ± 9</td>
<td></td>
</tr>
</tbody>
</table>

VII.III Results

We present the results for the rest of the equations in Fig. 3, Fig. 4, Fig. 5, Fig. 6 for the cosmological models Dark Parametric Energy with homoscedastic and heteroscedastic variance, Quintessence and Hu-Sawicky, respectively. And for the Cocoa fermentation model in Fig. 7.
Figure 3: Dark Parametric Energy with homoscedastic variance solutions

Figure 4: Dark Parametric Energy with heteroscedastic variance solutions

Figure 5: Quintessence solutions
Figure 6: Hu-Sawicky solutions
Figure 7: Cocoa fermentation solutions