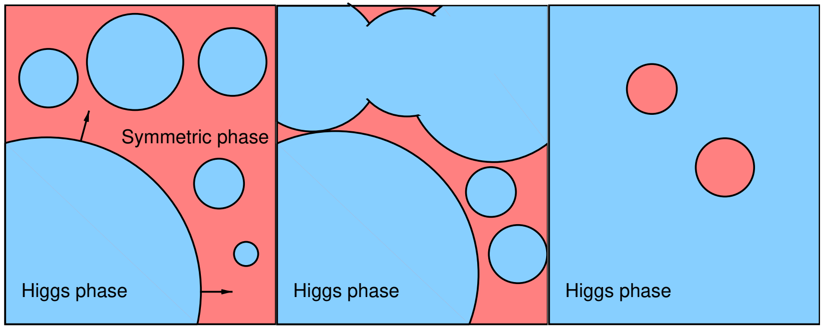
Self-similar hydrodynamics of first-order phase transitions in the early universe

Mika Mäki

2023-04-13

Bubbles are hydrodynamics

- How to compute the properties of a first-order phase transition bubble?
  - Hydrodynamics
- From the energy-momentum tensor to
  - Wave equation
  - Bubble wall junction conditions
  - Continuity equations, aka. hydrodynamic equations



Dimensionality of the problem

- Self-similarity
  - Friction results in a constant wall speed

    → As the bubble expands, its relative shape stays the same
    → Time-independent solution
- Spherical symmetry
- 3+1 dimensional problem reduces to time-independent 1D

Energy-momentum tensor

- In Minkowski space and Cartesian coordinates

$$T_{\mu\nu} = \begin{bmatrix} e & -q_1 & -q_2 & -q_3 \\ -q_1 & p+\Pi_{11} & \Pi_{12} & \Pi_{13} \\ -q_2 & \Pi_{12} & p+\Pi_{22} & \Pi_{23} \\ -q_3 & \Pi_{13} & \Pi_{23} & p+\Pi_{33} \end{bmatrix}$$

$$p = \frac{1}{3} T^i_i$$

- Ideal fluid

$$T_{\mu\nu} = \begin{bmatrix} e & 0 & 0 & 0 \\ 0 & p & 0 & 0 \\ 0 & 0 & p & 0 \\ 0 & 0 & 0 & p \end{bmatrix}$$

$$T_f^{\mu\nu} = (e+p)u^\mu u^\nu + p g^{\mu\nu}$$

Wave equation

- Constant background space-time
  → energy-momentum conservation $$\nabla_\mu T^{\mu\nu} = 0$$

- Energy-momentum tensor of an ideal fluid
$$T_f^{\mu\nu} = (e+p)u^\mu u^\nu + pg^{\mu\nu}$$

- For a one-dimensional flow in Cartesian coordinates
$$\partial_t\left[(e+pv^2)\gamma^2\right] + \partial_x\left[(e+p)\gamma^2 v\right] = 0,$$
$$\partial_t\left[(e+p)\gamma^2 v\right] + \partial_x\left[(ev^2+p)\gamma^2\right] = 0$$

- Frst-order perturbation → wave equation with speed of sound:
$$\partial_t^2(\delta e) - \frac{\delta p}{\delta e}\partial_x^2(\delta e) = 0 \qquad c_s^2 \equiv \frac{dp}{de} = \frac{dp/dT}{de/dT}$$

Phase boundary

- Energy-momentum conservation

$$\nabla_\mu T^{\mu\nu} = 0 \quad \Rightarrow \quad \partial_z T^{zz} = \partial_z T^{z0} = 0$$

- Inserting ideal fluid: $T_f^{\mu\nu} = (e+p)u^\mu u^\nu + p g^{\mu\nu}$

$$\theta \equiv \frac{1}{4}(e-3p)$$

$$w_- \tilde{\gamma}_-^2 \tilde{v}_- = w_+ \tilde{\gamma}_+^2 \tilde{v}_+,$$

$$w_- \tilde{\gamma}_-^2 \tilde{v}_-^2 + p_- = w_+ \tilde{\gamma}_+^2 \tilde{v}_+^2 + p_+$$

$$\Rightarrow \quad \tilde{v}_+ \tilde{v}_- = \frac{p_+ - p_-}{e_+ - e_-}, \qquad \frac{\tilde{v}_+}{\tilde{v}_-} = \frac{e_- + p_+}{e_+ + p_-}$$

$$\Delta\theta \equiv \theta_+(w_+) - \theta_-(w_-)$$

$$\alpha_+ \equiv \frac{4\Delta\theta}{3w_+}$$

$$\alpha_n \equiv \frac{4}{3}\frac{\theta_+(w_n) - \theta_-(w_n)}{w_n}$$

- By defining new variables

$$\tilde{v}_+ \tilde{v}_- = \frac{1-(1-3\alpha_+)r}{3-3(1+\alpha_+)r}$$

$$\frac{\tilde{v}_+}{\tilde{v}_-} = \frac{3+(1-3\alpha_+)r}{1+3(1+\alpha_+)r}$$

$$\Rightarrow \quad \tilde{v}_+ = \frac{1}{2(1+\alpha_+)}\left[\left(\frac{1}{3\tilde{v}_-} + \tilde{v}_-\right) \pm \sqrt{\left(\frac{1}{3\tilde{v}_-} - \tilde{v}_-\right)^2 + 4\alpha_+^2 + \frac{8}{3}\alpha_+}\right],$$

$$\tilde{v}_- = \frac{1}{2}\left[\left((1+\alpha_+)\tilde{v}_+ + \frac{1-3\alpha_+}{3\tilde{v}_+}\right) \pm \sqrt{\left((1+\alpha_+)\tilde{v}_+ + \frac{1-3\alpha_+}{3\tilde{v}_+}\right)^2 - \frac{4}{3}}\right].$$

Hydrodynamic equations

- Energy-momentum conservation

$$\nabla_\mu T^{\mu\nu} = 0$$

- Projection
  → hydrodynamic equations

$$0 = u_\mu \partial_\nu T^{\mu\nu} = -\partial_\mu (w u^\mu) + u^\mu \partial_\mu p,$$

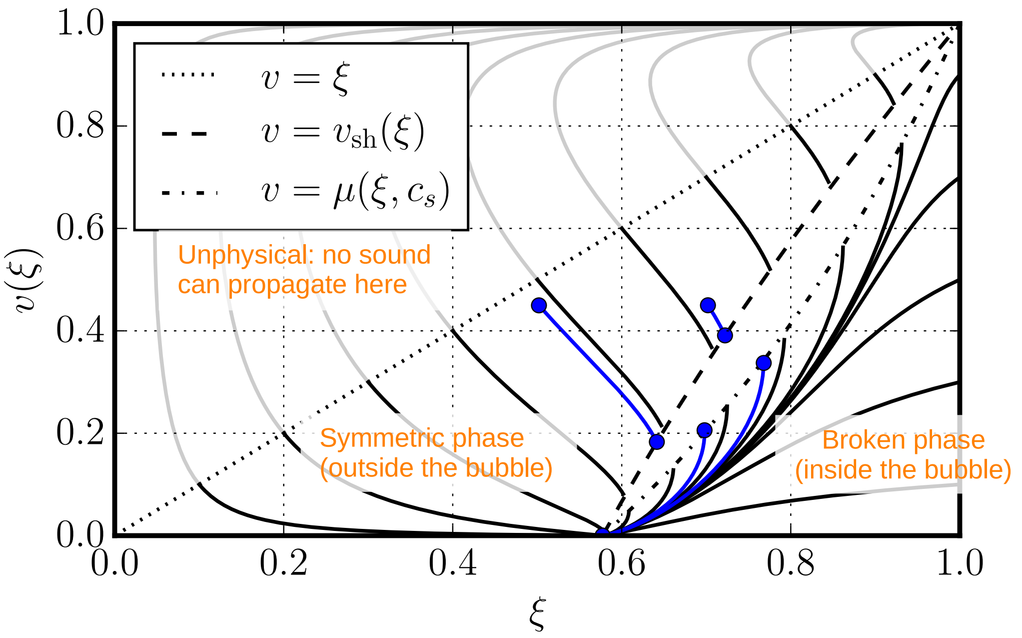
$$0 = \bar{u}_\mu \partial_\nu T^{\mu\nu} = w \bar{u}^\nu u^\mu \partial_\mu u_\nu + \bar{u}^\mu \partial_\mu p.$$

- Using self-similarity $\xi = \frac{r}{t}$

$$\frac{d\xi}{d\tau} = \xi \left[(\xi - v)^2 - c_s^2 (1 - \xi v)^2 \right],$$

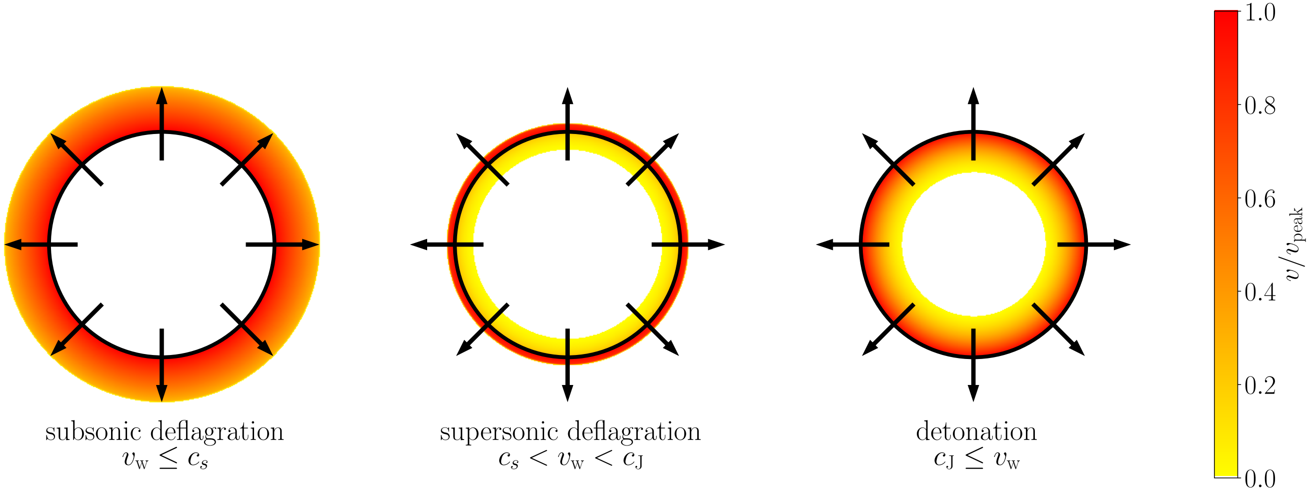
$$\frac{dv}{d\tau} = 2 v c_s^2 (1 - v^2)(1 - \xi v),$$

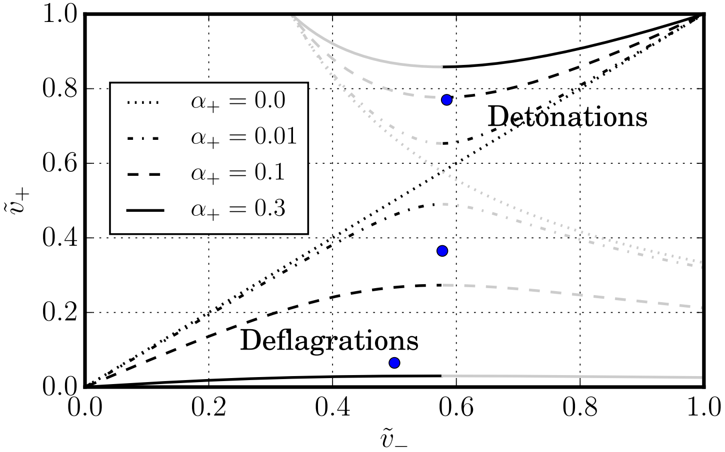
$$\frac{dw}{d\tau} = w \left(1 + \frac{1}{c_s^2} \right) \gamma^2 \mu \frac{dv}{d\tau}.$$



Fluid shells

- Three types of solutions, determined by
  - Wall velocity v_{wall}
  - Transition strength α_n





Black: bubble wall / phase boundary
Colour: velocity of moving plasma

Deriving the generic equation of state

$$T^{\mu\nu}(x) = \int \frac{d^3p}{(2\pi)^3} \frac{p^\mu p^\nu}{p^0} f(\vec{p}, x)$$

Pressure from momentum integral

$$P = \frac{1}{3} T_i^{\ i}$$

$$= \frac{1}{3} \int \frac{d^3p}{(2\pi)^3} \frac{p^i p_i}{p^0} f(\vec{p}, x)$$

$$p^0 = E$$

$$= \frac{1}{3} \int \frac{d^3p}{(2\pi)^3} \frac{|\vec{p}|^2}{E} f(\vec{p}, x)$$

Bose-Einstein

$$f(\vec{p}) = \frac{1}{e^{\frac{E-\mu}{T}} - 1}$$

ultrarelativistic

$$E(\vec{p}) = \sqrt{\vec{p}^2 + m^2} \approx |p|, \quad \mu = 0$$

$$= \frac{1}{3} \int \frac{d^3p}{(2\pi)^3} \frac{|\vec{p}|^2}{|p|} \frac{1}{e^{\frac{p}{T}} - 1}$$

3D integral to 1D

$$A(r) = 4\pi r^2$$

$$= \frac{1}{3} \int_0^\infty \frac{4\pi p^2 dp}{(2\pi)^3} \, p \, \frac{1}{e^{\frac{p}{T}} - 1}$$

Mathematics

$$\int_0^\infty \frac{x^n}{e^x - 1} = \Gamma(n+1)\,\zeta(n+1)$$

$$= \frac{1}{6\pi^2} \int_0^\infty \frac{p^3 dp}{e^{\frac{p}{T}} - 1}$$

$$= \frac{1}{6\pi^2} \int_0^\infty \frac{(\frac{p}{T})^3 d(\frac{p}{T})}{e^{\frac{p}{T}} - 1} T^4$$

$$= \frac{1}{6\pi^2} \Gamma(4)\,\zeta(4)\, T^4$$

$$= \frac{1}{\cancel{6}\pi^2} \cancel{6} \frac{\pi^4}{90} T^4$$

$$= \frac{\pi^2}{90} T^4$$

Accounting for multiple fields and an external potential

$$\rightarrow P(T, \phi) = \frac{\pi^2}{90} g_P(T) T^4 - V(T, \phi)$$

Bag model: the simplest model

- Equation of state: $p(T, \phi)$

  $$p_s = a_s T^4 - V_s$$

  $$p_b = a_b T^4$$

- The rest can be deduced with thermodynamics
  - Enthalpy density w
  - Energy density e
  - Entropy density s
  - Sound speed c_s

$$w \equiv T \frac{\partial p}{\partial T} = e + p = Ts$$

$$c_s^2 \equiv \left(\frac{\partial p}{\partial e}\right)_s = \frac{1}{3}$$

Beyond the bag model

- Non-constant degrees of freedom → Bag model assumptions broken
  - Different equations of state for each phase
  - Different sound speeds, possibly temperature-dependent: $c_s(T,\phi)$
- Computing the velocity profile becomes more difficult
  - Sound speed may change at each point

    → No analytical shortcuts in finding the solution
- Next approximation: Constant sound speed model
  - Different but constant speed of sound in each phase

$$p_s = a_s T^\mu - V_s \qquad \mu = 1 + \frac{1}{c_{s,s}^2}$$

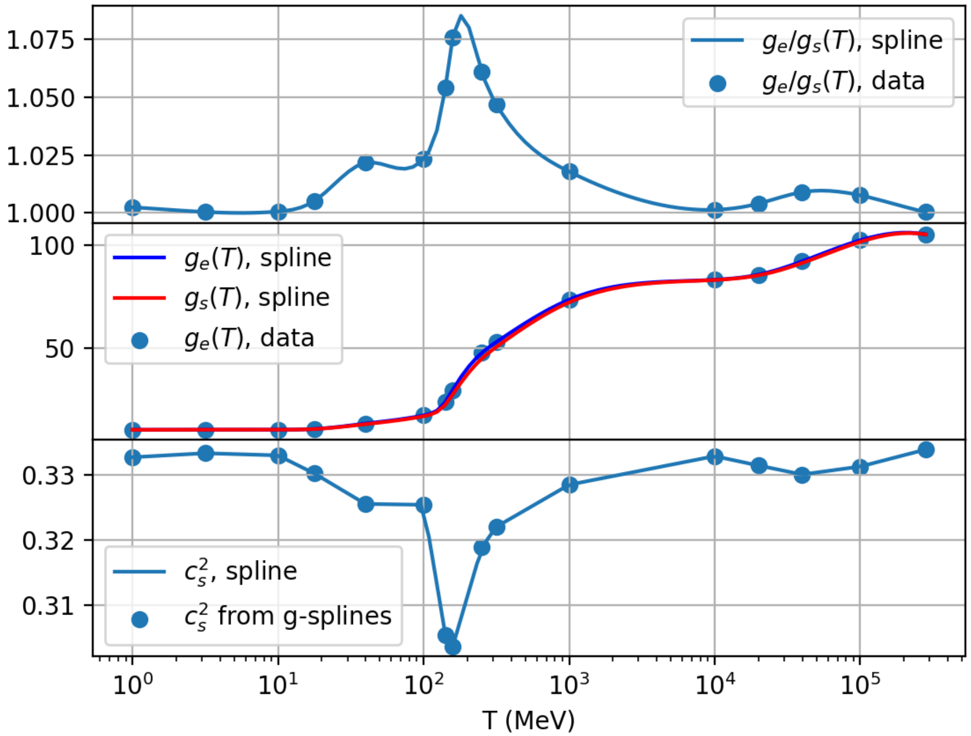
$$p_b = a_b T^\nu \qquad \nu = 1 + \frac{1}{c_{s,b}^2}$$

Goal: Equation of state from an arbitrary model

- Example: Standard Model
- Fluid properties depend on
  - Temperature T
  - Phase ϕ
- Arbitrary models can be tested, when $g_{\text{eff}}(T,\phi)$ is given

$$e(T,\phi) = \frac{\pi^2}{30} g_e(T,\phi) T^4$$

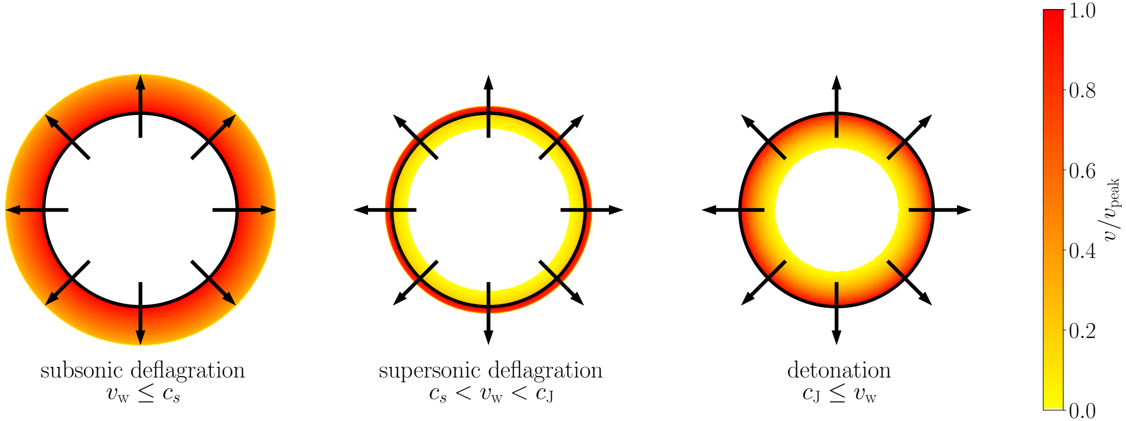
$$s(T,\phi) = \frac{2\pi^2}{45} g_s(T,\phi) T^3$$



Data from Borsanyi et al., 2016

Old algorithm

- Find the α_+ for which α_n is correct
- Compute fluid speeds at the wall from α_+
- Deflagrations and hybrids: integrate forward to shock
- Detonations and hybrids: integrate backward to sound speed (where v=0)



$$\theta \equiv \frac{1}{4}(e - 3p)$$

$$\Delta\theta \equiv \theta_+(w_+) - \theta_-(w_-)$$

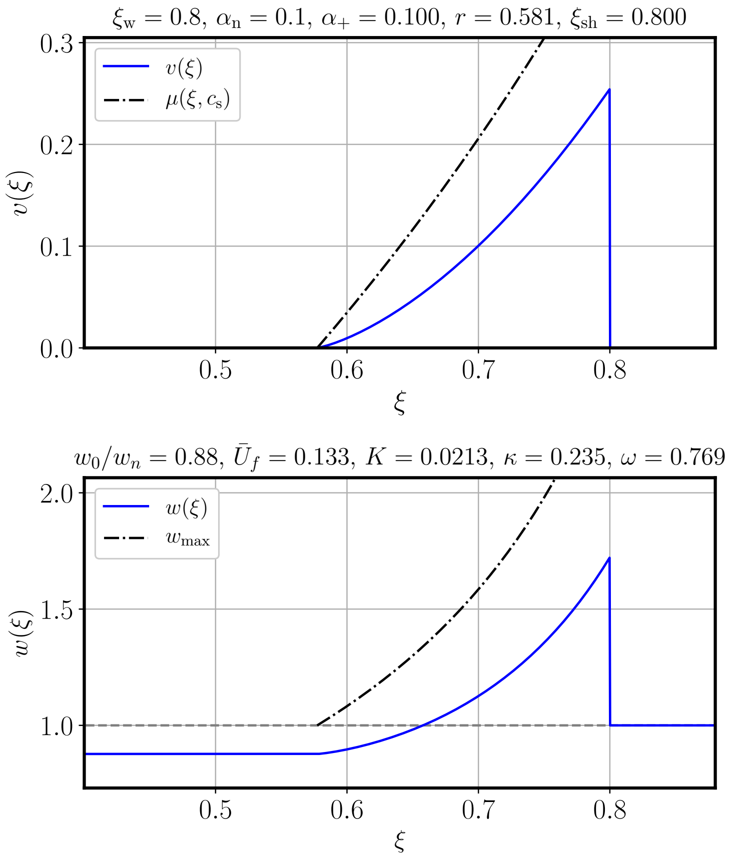
$$\alpha_+ \equiv \frac{4\Delta\theta}{3w_+}$$

$$\alpha_n \equiv \frac{4}{3}\frac{\theta_+(w_n) - \theta_-(w_n)}{w_n}$$

Fluid shell solver algorithm

Detonations

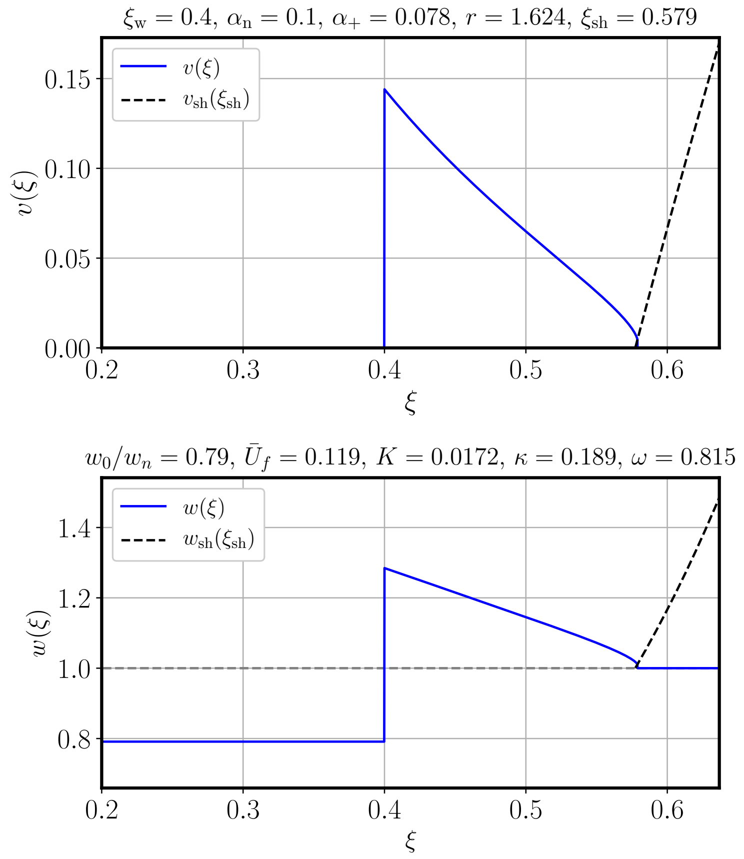
- Start from the known conditions outside the wall
- Solve boundary conditions at the wall
  - SciPy fsolve (MINPACK hybrd, hybrj)
- Integrate to v=0
  - SciPy odeint, SciPy solve_ivp, NumbaLSODA
- Extend to the center of the bubble



Fluid shell solver algorithm

Subsonic deflagrations

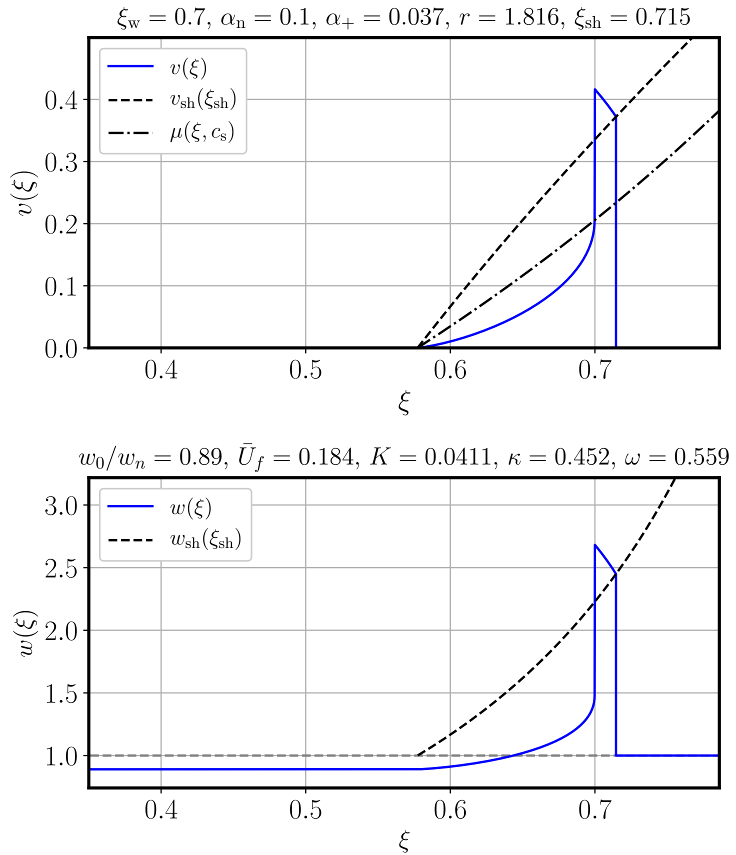
- Guess an enthalpy within the bubble
- Solve boundary conditions at the wall
  - A solver within a solver
- Integrate to the shock
  - The curve turns backwards at v > 0
    → outside can only be reached with a shock
- Solve boundary conditions at the shock
- Check if w=wn
- If not, change the enthalpy guess



Fluid shell solver algorithm

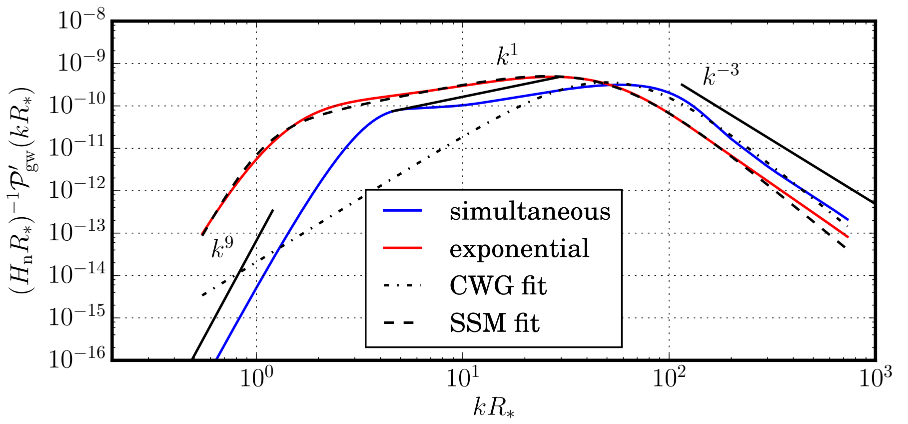
Hybrids (supersonic deflagrations)

- As for subsonic deflagrations
  - Guess an enthalpy behind the wall, $\tilde{v}_- = c_{s-}(w_-)$
  - Solve boundary conditions at the wall
  - Integrate to the shock
  - Solve boundary conditions at the shock
  - Check if w=wn
  - If not, change the enthalpy guess
- Integrate from the wall to v=0



GW production

- Sine transform
  → velocity power spectrum
- Convolution with the nucleation rate function
  → overall velocity power spectrum
- Convolution with a Green's function
  → GW power spectrum

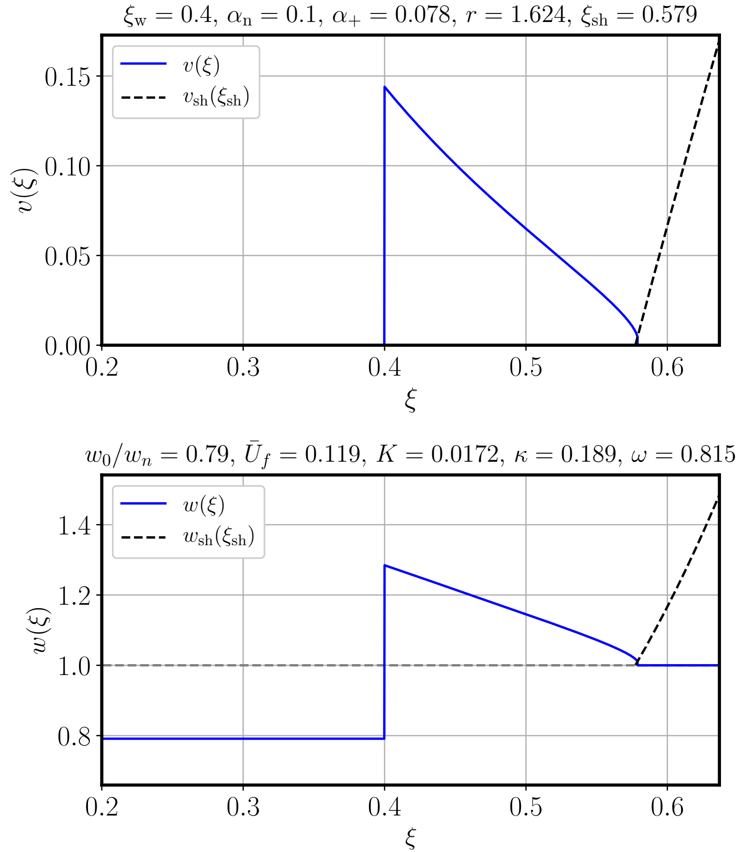


GW power spectrum is characterized by five key parameters

- GW power spectrum is characterized by
  - Nucleation temperature T_n
  - Phase transition strength at the nucleation temperature α_n
  - Bubble wall speed $v_{\rm wall}$
  - Transition rate parameter β
  - Sound speed $c_s(T,\phi)$
- Initial analysis: simple toy models
- **Goal of the thesis: arbitrary model from particle physics parameters**

Numerical tricks & bugs

- The solver converges to the correct solution only when the starting guess is close to the correct solution
  - Solution: use the old bag model solver as a starting guess
- Insufficient accuracy of the ODE integrator at the shock
  - Solution: use analytical knowledge about the shape of the curve to accept the nearest point
- Numerical equation group solver gets stuck
  - Solution: vary the initial guesses

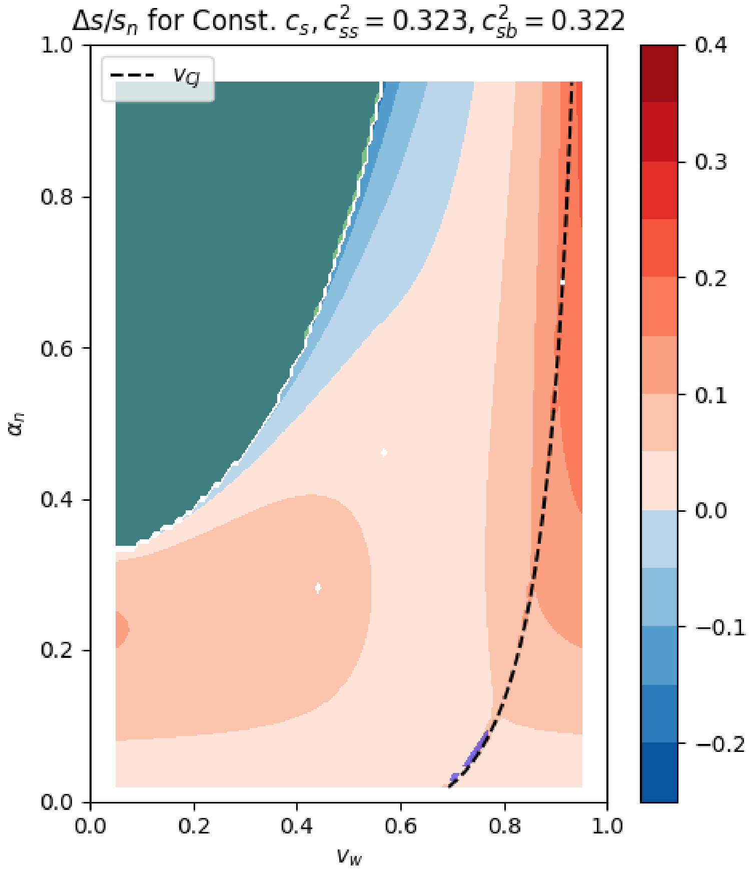


Numerical solving takes time

- Testing with e.g. 100 values for v_{wall} and α_n

  → 10 000 bubbles! And multiple full iterations for each until the solution is found

- Python is single-threaded
  - Solution: parallelism with concurrent.futures
- Laptops are slow, but development requires quick feedback
  - Solution: CFT remote workstation, SSH connection, IDE integration



Speedup: Numba

- Python is an interpreted language
  - Orders of magnitude slower than compiled languages such as C, C++ and FORTRAN
  - Most of the heavy mathematics is done inside Numpy, which is written in C
- Numba can compile [a subset of] Python code to native binary
  - Using the LLVM compiler framework
  - Speed comparable to C and FORTRAN
  - Support for GPUs: CUDA & ROCm
- Python is single-threaded (GIL), albeit with concurrency, but Numba can unlock true parallelism within a single process

Speedup example with Numba

- Add JIT decorator
- Replace unsupported features with simpler code or split the unsupported parts to another function
- (Restructure the function to make possible parallelism explicit)

```python
@numba.njit(parallel=True)
def sin_transform_core(t: np.ndarray, f: np.ndarray, freq: np.ndarray) -> np.ndarray:
    integral = np.zeros_like(freq)
    for i in numba.prange(freq.size):
        integrand = f * np.sin(freq[i] * t)
        integral[i] = np.trapz(integrand, t)
    return integral
```

$$\hat{f}(\omega) = \int_{t_{\min}}^{t_{\max}} f(t)\sin(\omega t)dt$$

Summary

- Hydrodynamics is based on energy-momentum conservation
- Solving a bubble starts with the equation of state

  + equation solving & ODE integration → fluid velocity profile
  + sine transform & convolutions → GW power spectrum
- PTtools now supports arbitary equations of state
  - Will be published soon

Thank you!

Sources

- M. Hindmarsh, M. Lüben, J. Lumma, and M. Pauly, “Phase transitions in the early universe,” SciPost Phys. Lect. Notes, Feb. 2021, doi: 10.21468/SciPostPhysLectNotes.24.
- Hindmarsh, Mark, and Mulham Hijazi. “Gravitational Waves from First Order Cosmological Phase Transitions in the Sound Shell Model.” Journal of Cosmology and Astroparticle Physics 2019, Dec. 2019, doi: 10.1088/1475-7516/2019/12/062.
- Caprini et al. “Detecting Gravitational Waves from Cosmological Phase Transitions with LISA: An Update.” Journal of Cosmology and Astroparticle Physics 2020, Mar. 2020, doi: 10.1088/1475-7516/2020/03/024.
- Espinosa et al. “Energy Budget of Cosmological First-Order Phase Transitions.” Journal of Cosmology and Astroparticle Physics 2010, Jun. 2010, doi: 10.1088/1475-7516/2010/06/028.
- Giese et al., “Model-Independent Energy Budget for LISA.” Journal of Cosmology and Astroparticle Physics, Jan. 2021, doi: 10.1088/1475-7516/2021/01/072.
- Borsanyi, Sz, Z. Fodor, K. H. Kampert, S. D. Katz, T. Kawanai, T. G. Kovacs, S. W. Mages, et al. “Lattice QCD for Cosmology.”, Jun. 2016, ArXiv: 1606.07494