

EDF R&D	Validation of <i>Code_Saturne</i> 6.0.2 (Appendix 1)	???-???-2019-???-EN Version 1.0
---------	---	--

Sommaire / Summary

1	Lagrangian Point Source Dispersion	1
1.1	Description of the test case	1
1.1.1	Introduction	1
1.1.2	Theory	1
1.1.3	References	6
1.2	Numerical set up	6
1.2.1	Computational domain	6
1.2.2	Physical modeling for the fluid flow	6
1.2.3	Physical modelling for the particles	7
1.3	Results	8
1.4	Conclusions	12

EDF R&D	Validation of <i>Code_Saturne</i> 6.0.2 (Appendix 1)	???-???-2019-???-EN Version 1.0
---------	---	--

1 Lagrangian Point Source Dispersion

Author: Christophe Henry, (INRIA Sophia Antipolis), Martin Ferrand, RENUA
update: 27/07/20

Last

Key-words: 3D, Lagrangian, Dispersion

This test case corresponds to a verification case for the Lagrangian module in *Code_Saturne*. It consists of point source dispersion.

1.1 Description of the test case

1.1.1 Introduction

This test case consists in a fluid at rest in a cubic mesh (no flow) inside which particles are initially injected in the centre of the domain.

To verify the present algorithm for the treatment of the equations of particle motion, numerical results are compared to analytical results in ideal cases that correspond to limiting cases (with constant coefficients).

This test case consists in an isotropic case as the one described in [1] and [2]: constant coefficients $C_i(t, \mathbf{x}_p) = 0$, $\mathbf{A}_i(t, \mathbf{x}_p) = 0$ and initial conditions $x_p(0) = U_p(0) = U_s(0) = 0$. In that case, the system simplifies to

$$\begin{cases} dx_p = U_p dt \\ dU_p = \frac{1}{\tau_p}(U_s - U_p) dt \\ dU_s = -\frac{1}{T}U_s dt + \sigma dW(t), \end{cases} \quad (1.1)$$

and the solution of the system is

$$\begin{cases} x_p = \Omega(t) \\ U_p = \Gamma(t) \\ U_s = \gamma(t), \end{cases} \quad (1.2)$$

The present test case was updated compared to the one detailed in [2]. Previously, the particles were injected at an inlet face and then displaced to the centre. This led to having a combination of 5 symmetry boundary conditions and one inlet boundary condition. Here, the particles are injected directly in the cell, which makes it possible to impose symmetry boundary conditions on all the cell faces and leads to slightly better results.

1.1.2 Theory

The dynamics of discrete particles and the corresponding system of SDEs has been described in details in [3] and only the main features are recalled here. The system of SDEs describing the dynamics of the

EDF R&D	Validation of <i>Code_Saturne</i> 6.0.2 (Appendix 1)	????-????-2019-????-EN Version 1.0
---------	---	---

discrete particles reads

$$\begin{cases} dx_{p,i}(t) = U_{p,i} dt, \\ dU_{p,i}(t) = \frac{1}{\tau_p} (U_{s,i} - U_{p,i}) dt + \mathcal{A}_i dt, \\ dU_{s,i}(t) = -\frac{1}{T_{L,i}^*} U_{s,i} dt + C_i dt + \sum_j B_{ij} dW_j(t), \end{cases} \quad (1.3)$$

where C_i is a term that includes all mean contributions: the mean pressure gradient, $-(\partial\langle P\rangle/\partial x_i)/\rho_f$, the mean drift term, $(\langle U_{p,j}\rangle - \langle U_j\rangle)(\partial\langle U_i\rangle/\partial x_j)$, and the mean part of the return-to-equilibrium term, $\langle U_i\rangle/T_{L,i}^*$. \mathcal{A}_i is an acceleration (gravity in the present work, but it can be extended for practical reasons to the case of other external force fields).

The weak numerical schemes, with the required features, are developed based on the analytical solution to Eqs. (1.3) *with constant coefficients* (independent of time), the main idea being to derive a numerical scheme by freezing the coefficients on the integration intervals. This methodology ensures *stability* and *consistency with all limit systems*:

- stability because the form of the equations gives analytical solutions with exponentials of the type $\exp(-\Delta t/T)$ where T is one of the characteristic timescales (τ_p and $T_{L,i}^*$),
- consistency with all limit systems by construction, since the schemes are based on an analytical solution.

Different techniques shall be used to derive first and second-order (in time) schemes from the analytical solutions with constant coefficients. A first-order scheme can be obtained by computing, at each time step, the variables on the basis of the analytical solutions (all coefficients are frozen at the beginning of the integration interval), i.e. a numerical scheme of the *Euler* kind is obtained. A second-order scheme can be derived by resorting to a predictor-corrector technique where the prediction step is the first-order scheme.

Analytical solution Before presenting the weak numerical schemes, it is a prerequisite to give the analytical solutions to system (1.3), with constant coefficients (in time). These solutions are obtained by resorting to Itô's calculus in combination with the method of the variation of the constant. For instance, for the fluid velocity seen, one seeks a solution of the form $U_{s,i}(t) = H_i(t) \exp(-t/T_i)$, where $H_i(t)$ is a stochastic process defined by (*from now on the notation is slightly changed: $T_{L,i}^*$ is noted T_i for the sake of clarity in the complex formulae to come*)

$$dH_i(t) = \exp(t/T_i)[C_i dt + \check{B}_i dW_i(t)], \quad (1.4)$$

that is, by integration on a time interval $[t_0, t]$ ($\Delta t = t - t_0$),

$$\begin{aligned} U_{s,i}(t) &= U_{s,i}(t_0) \exp(-\Delta t/T_i) + C_i T_i [1 - \exp(-\Delta t/T_i)] \\ &\quad + \check{B}_i \exp(-t/T_i) \int_{t_0}^t \exp(s/T_i) dW_i(s), \end{aligned} \quad (1.5)$$

where $\check{B}_i = B_{ii}$ since B_{ij} is a diagonal matrix). By proceeding in the same way for the other equations (position and velocity), the analytical solution is obtained for the entire system, cf. Table 1.1. The three stochastic integrals, Eqs. (1.9) to (1.11) in Table 1.1, are centred Gaussian processes. These integrals are defined implicitly, but they can be simplified by integration by parts, cf. Table 1.1.

Weak first-order scheme From the analytical solutions of this system assuming constant coefficients, the weak-first order scheme is extracted. The equations are described in Table 1.2.

Weak second-order scheme The weak second-order scheme consists in a correction step for the particle velocity and the velocity of the fluid seen as described in Table 1.3:

EDF R&D	Validation of <i>Code_Saturne</i> 6.0.2 (Appendix 1)	???-???-2019-???-EN Version 1.0
---------	---	--

Table 1.1: Analytical solutions to system (1.3) for time-independent coefficients.

$$\begin{aligned}
x_{p,i}(t) &= x_{p,i}(t_0) + U_{p,i}(t_0)\tau_p[1 - \exp(-\Delta t/\tau_p)] + U_{s,i}(t_0)\theta_i\{T_i[1 - \exp(-\Delta t/T_i)] \\
&\quad + \tau_p[\exp(-\Delta t/\tau_p) - 1]\} + [C_i T_i]\{\Delta t - \tau_p[1 - \exp(-\Delta t/\tau_p)] \\
&\quad - \theta_i(T_i[1 - \exp(-\Delta t/T_i)] + \tau_p[\exp(-\Delta t/\tau_p) - 1])\} + \Omega_i(t) \\
&\quad \text{with } \theta_i = T_i/(T_i - \tau_p)
\end{aligned} \tag{1.6}$$

$$\begin{aligned}
U_{p,i}(t) &= U_{p,i}(t_0)\exp(-\Delta t/\tau_p) + U_{s,i}(t_0)\theta_i[\exp(-\Delta t/T_i) - \exp(-\Delta t/\tau_p)] \\
&\quad + [C_i T_i]\{[1 - \exp(-\Delta t/\tau_p)] - \theta_i[\exp(-\Delta t/T_i) - \exp(-\Delta t/\tau_p)]\} \\
&\quad + \Gamma_i(t)
\end{aligned} \tag{1.7}$$

$$U_{s,i}(t) = U_{s,i}(t_0)\exp(-\Delta t/T_i) + C_i T_i[1 - \exp(-\Delta t/T_i)] + \gamma_i(t) \tag{1.8}$$

The stochastic integrals $\gamma_i(t)$, $\Gamma_i(t)$, $\Omega_i(t)$ are given by:

$$\gamma_i(t) = \check{B}_i \exp(-t/T_i) \int_{t_0}^t \exp(s/T_i) dW_i(s), \tag{1.9}$$

$$\Gamma_i(t) = \frac{1}{\tau_p} \exp(-t/\tau_p) \int_{t_0}^t \exp(s/\tau_p) \gamma_i(s) ds, \tag{1.10}$$

$$\Omega_i(t) = \int_{t_0}^t \Gamma_i(s) ds. \tag{1.11}$$

By resorting to stochastic integration by parts, $\gamma_i(t)$, $\Gamma_i(t)$, $\Omega_i(t)$ can be written:

$$\gamma_i(t) = \check{B}_i \exp(-t/T_i) I_{1,i}, \tag{1.12}$$

$$\Gamma_i(t) = \theta_i \check{B}_i [\exp(-t/T_i) I_{1,i} - \exp(-t/\tau_p) I_{2,i}], \tag{1.13}$$

$$\begin{aligned}
\Omega_i(t) &= \theta_i \check{B}_i \{(T_i - \tau_p) I_{3,i} \\
&\quad - [T_i \exp(-t/T_i) I_{1,i} - \tau_p \exp(-t/\tau_p) I_{2,i}]\},
\end{aligned} \tag{1.14}$$

$$\begin{aligned}
\text{with } I_{1,i} &= \int_{t_0}^t \exp(s/T_i) dW_i(s), \quad I_{2,i} = \int_{t_0}^t \exp(s/\tau_p) dW_i(s) \\
&\quad \text{and } I_{3,i} = \int_{t_0}^t dW_i(s).
\end{aligned}$$

EDF R&D	Validation of Code_Saturne 6.0.2 (Appendix 1)	????-????-2019-????-EN Version 1.0
---------	--	---

Table 1.2: Weak first-order scheme (Euler scheme)

Numerical integration of the system:

$$\begin{aligned}
x_{p,i}^{n+1} &= x_{p,i}^n + A_1 U_{p,i}^n + B_1 U_{s,i}^n + C_1 [T_i^n C_i^n] + \Gamma_i^n, \\
U_{p,i}^{n+1} &= U_{p,i}^n \exp(-\Delta t / \tau_p^n) + D_1 U_{s,i}^n + [T_i^n C_i^n] (E_1 - D_1) + \Omega_i^n, \\
U_{s,i}^{n+1} &= U_{s,i}^n \exp(-\Delta t / T_i^n) + [T_i^n C_i^n] [1 - \exp(-\Delta t / T_i^n)] + \gamma_i^n.
\end{aligned}$$

The coefficients A_1 , B_1 , C_1 , D_1 and E_1 are given by:

$$\begin{aligned}
A_1 &= \tau_p^n [1 - \exp(-\Delta t / \tau_p^n)], \\
B_1 &= \theta_i^n [T_i^n (1 - \exp(-\Delta t / T_i^n) - A_1)] \quad \text{with} \quad \theta_i^n = T_i^n / (T_i^n - \tau_p^n), \\
C_1 &= \Delta t - A_1 - B_1, \\
D_1 &= \theta_i^n [\exp(-\Delta t / T_i^n) - \exp(-\Delta t / \tau_p^n)], \\
E_1 &= 1 - \exp(-\Delta t / \tau_p^n).
\end{aligned}$$

The stochastic integrals γ_i^n , Ω_i^n , Γ_i^n are simulated by:

$$\begin{aligned}
\gamma_i^n &= P_{11} \mathcal{G}_{1,i}, \\
\Omega_i^n &= P_{21} \mathcal{G}_{1,i} + P_{22} \mathcal{G}_{2,i}, \\
\Gamma_i^n &= P_{31} \mathcal{G}_{1,i} + P_{32} \mathcal{G}_{2,i} + P_{33} \mathcal{G}_{3,i},
\end{aligned}$$

where $\mathcal{G}_{1,i}$, $\mathcal{G}_{2,i}$, $\mathcal{G}_{3,i}$ are independent $\mathcal{N}(0, 1)$ random variables.

The coefficients P_{11} , P_{21} , P_{22} , P_{31} , P_{32} , P_{33} are defined as:

$$\begin{aligned}
P_{11} &= \sqrt{\langle (\gamma_i^n)^2 \rangle}, \\
P_{21} &= \frac{\langle \Gamma_i^n \gamma_i^n \rangle}{\sqrt{\langle (\gamma_i^n)^2 \rangle}}, \quad P_{22} = \sqrt{\langle (\Gamma_i^n)^2 \rangle - \frac{\langle \Gamma_i^n \gamma_i^n \rangle^2}{\langle (\gamma_i^n)^2 \rangle}}, \\
P_{31} &= \frac{\langle \Omega_i^n \gamma_i^n \rangle}{\sqrt{\langle (\gamma_i^n)^2 \rangle}}, \quad P_{32} = \frac{1}{P_{22}} (\langle \Omega_i^n \Gamma_i^n \rangle - P_{21} P_{31}), \quad P_{33} = \sqrt{\langle (\Omega_i^n)^2 \rangle - P_{31}^2 - P_{32}^2}.
\end{aligned}$$

EDF R&D	Validation of <i>Code_Saturne</i> 6.0.2 (Appendix 1)	????-????-2019-????-EN Version 1.0
---------	---	---

Table 1.3: Weak second-order scheme

Prediction step: Euler scheme, see Table 1.2 (predicted values noted with \sim).

Correction step:

$$\begin{aligned}
U_{p,i}^{n+1} &= \frac{1}{2} U_{p,i}^n \exp(-\Delta t / \tau_p^n) + \frac{1}{2} U_{p,i}^n \exp(-\Delta t / \tilde{\tau}_p^{n+1}) \\
&\quad + \frac{1}{2} U_{s,i}^n C_{2c}(\tau_p^n, T_i^n) + \frac{1}{2} U_{s,i}^n C_{2c}(\tilde{\tau}_p^{n+1}, \tilde{T}_i^{n+1}) \\
&\quad + A_{2c}(\tau_p^n, T_i^n) [T_i^n C_i^n] + B_{2c}(\tilde{\tau}_p^{n+1}, \tilde{T}_i^{n+1}) [\tilde{T}_i^{n+1} C_i^{n+1}] \\
&\quad + A_2(\Delta t, \tau_p^n) [\tau_p^n \mathcal{A}_i^n] + B_2(\Delta t, \tilde{\tau}_p^{n+1}) [\tilde{\tau}_p^{n+1} \mathcal{A}_i^{n+1}] + \tilde{\Gamma}_i^{n+1}, \\
U_{s,i}^{n+1} &= \frac{1}{2} U_{s,i}^n \exp(-\Delta t / T_i^n) + \frac{1}{2} U_{s,i}^n \exp(-\Delta t / \tilde{T}_i^{n+1}) + A_2(\Delta t, T_i^n) [T_i^n C_i^n] \\
&\quad + B_2(\Delta t, \tilde{T}_i^{n+1}) [\tilde{T}_i^{n+1} C_i^{n+1}] + \tilde{\gamma}_i^{n+1}.
\end{aligned}$$

The coefficients A_2 , B_2 , A_{2c} , B_{2c} et C_{2c} are defined as:

$$\begin{aligned}
A_2(\Delta t, x) &= -\exp(-\Delta t / x) + [1 - \exp(-\Delta t / x)][x / \Delta t], \\
B_2(\Delta t, x) &= 1 - [1 - \exp(-\Delta t / x)][x / \Delta t], \\
A_{2c}(x, y) &= -\exp(-\Delta t / x) + [(x + y) / \Delta t][1 - \exp(-\Delta t / x)] - (1 + y / \Delta t) C_{2c}(x, y), \\
B_{2c}(x, y) &= 1 - [(x + y) / \Delta t][1 - \exp(-\Delta t / x)] + (y / \Delta t) C_{2c}(x, y), \\
C_{2c}(x, y) &= [y / (y - x)][\exp(-\Delta t / y) - \exp(-\Delta t / x)].
\end{aligned}$$

The stochastic integrals $\tilde{\gamma}_i^{n+1}$ and $\tilde{\Gamma}_i^{n+1}$ are simulated as follows:

$$\begin{aligned}
\tilde{\gamma}_i^{n+1} &= \sqrt{\frac{[B_i^*]^2 \tilde{T}_i^{n+1}}{2}} [1 - \exp(-2\Delta t / \tilde{T}_i^{n+1})] \mathcal{G}_{1,i}, \\
\text{with } [1 - \exp(-2\Delta t / \tilde{T}_i^{n+1})] B_i^* &= A_2(2\Delta t, \tilde{T}_i^{n+1}) \sqrt{(\tilde{B}_i^n)^2 +} \\
&\quad B_2(2\Delta t, \tilde{T}_i^{n+1}) \sqrt{(\tilde{B}_i^{n+1})^2}. \\
\tilde{\Gamma}_i^{n+1} &= \frac{\langle \tilde{\Gamma}_i^{n+1} \tilde{\gamma}_i^{n+1} \rangle}{\langle (\tilde{\gamma}_i^{n+1})^2 \rangle} \tilde{\gamma}_i^{n+1} + \sqrt{\frac{\langle (\tilde{\Gamma}_i^{n+1})^2 \rangle - \frac{\langle \tilde{\Gamma}_i^{n+1} \tilde{\gamma}_i^{n+1} \rangle^2}{\langle (\tilde{\gamma}_i^{n+1})^2 \rangle}}{\langle (\tilde{\gamma}_i^{n+1})^2 \rangle}} \mathcal{G}_{2,i}
\end{aligned}$$

$$\text{with } \langle \tilde{\Gamma}_i^{n+1} \tilde{\gamma}_i^{n+1} \rangle = \langle \Gamma_i \gamma_i \rangle(\tau_p^n, \tilde{T}_i^{n+1}, B_i^*) \quad \text{and} \quad \langle (\tilde{\Gamma}_i^{n+1})^2 \rangle = \langle \Gamma_i^2 \rangle(\tau_p^n, \tilde{T}_i^{n+1}, B_i^*).$$

EDF R&D	Validation of <i>Code_Saturne</i> 6.0.2 (Appendix 1)	???-???-2019-???-EN Version 1.0
---------	---	--

1.1.3 References

- [1] J.-P. Minier, E. Peirano, and S. Chibbaro. Weak first and second order numerical schemes for stochastic differential equations appearing in lagrangian two-phase flow modelling. *Monte Carlo Methods and Applications*, 9(2):93–133, 2003.
- [2] C. HENRY, J. POZORSKI Modelling of agglomeration and deposition of colloidal particles carried by a flow: Fourth progress report. *EDF Report*, Report-EDF-09-2016, 2016.
- [3] E. Peirano, S. Chibbaro, J. Pozorski, and J.-P. Minier. Mean-field/PDF numerical approach for poly-dispersed turbulent two-phase flows. *Progress in Energy and Combustion Science*, 32(3):315–371, 2006.

1.2 Numerical set up

1.2.1 Computational domain

To perform a numerical simulation in such an ideal case, a very simple computational domain consisting of a single cubic cell (with a size of 1000 m) has been retained.

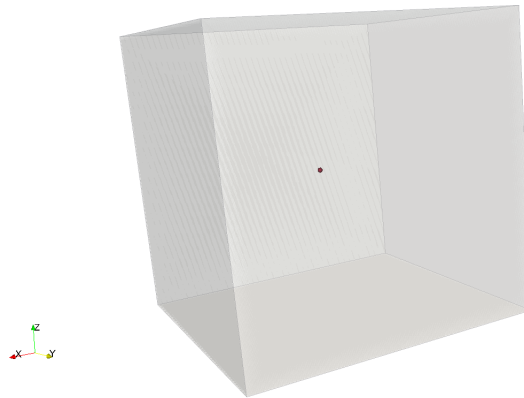


Figure 1.1: 3D view of the single cell used for the verification of the numerical scheme in ideal cases.

1.2.2 Physical modeling for the fluid flow

The fluid is at rest (no fluid motion). Since all parameters are fixed manually in the Lagrangian module of *Code_Saturne*, the simulation of the fluid phase is not primordial. Nevertheless, the following parameters have been retained to be able to run a two-phase flow simulation. The flow is isothermal and incompressible. Gravity has been neglected. The properties used for the liquid in the flow domain are summarised in Table 1.4:

U_{inlet}	μ_f	ρ_f
0 m s^{-1}	$1.002 \times 10^{-3} \text{ kg m}^{-1} \text{ s}^{-1}$	998 kg m^{-3}

Table 1.4: Fluid properties used in the simulation: inlet velocity U_{inlet} , Reynolds number Re , dynamic viscosity μ_f and density ρ_f .

A turbulence model has been activated to be able to use the particle turbulent dispersion model:

- Turbulence model: $k - \varepsilon$ (with initial values to get the desired Lagrangian times)
- Frozen dynamic field

EDF R&D	Validation of <i>Code_Saturne</i> 6.0.2 (Appendix 1)	???-???-2019-???-EN Version 1.0
---------	---	--

Initial conditions

The initial conditions are the following:

- reference and initial pressure: 101 325 Pa
- fluid velocity: 0.0 m s^{-1}
- temperature: 293.15 K

Boundary conditions

All the faces of the computational domain are defined as symmetry boundary conditions. The particles are injected directly at the centre of the computational domain.

1.2.3 Physical modelling for the particles

Injection Particles are injected in the domain. The properties used for the injected particles are:

- Particle diameter: $1 \times 10^{-3} \text{ m}$
- Monodispersed particles
- Particle density: 998 kg m^{-3}
- Frequency of injection: 0 (only initially)
- Number of particles in class: 20 000

Boundary condition The symmetry boundary conditions for particles have been set to

- symmetry (particles zero-flux)

Model for transport The CFD simulation with the injection of particles has been performed using the Lagrangian module in *Code_Saturne* and the following properties

- Eulerian-Lagrangian multi-phase treatment: one way coupling
- The continuous phase is a steady flow
- No additional models associated with the particles
- Integration for the stochastic differential equations: first-order scheme or second-order scheme
- Particle turbulent dispersion model: activated

Numerical parameters The two-phase flow simulation has been performed using the following numerical parameters:

- Calculation restart from the fluid flow simulation
- Constant time step: 0.001 s
- Number of iterations: 2000 or 4000

The subroutines `cs_user_initialization.c`, `cs_user_lagr_particle.c`, `cs_user_lagr_volume_conditions.c`, `cs_user_mesh-modify.c` are coded to set-up the test case.

- **`cs_user_initialization.c`** is used to initialize the turbulence k and ϵ from T_L and σ .
- **`cs_user_lagr_volume_conditions.c`** is used to set-up the particle class.
- **`cs_user_lagr_particle.c`** is used to set-up the parameter τ_p and to put all the particles at the middle of the domain (see also Figure 1.1).
- **`cs_user_mesh-modify.c`** is used to center and scale the domain to 1000 m.

The various cases studied are summarised in Table 1.5:

Case	τ_p (s)	T_L (s)	σ (m s ^{-3/2})
General case	10^{-1}	2×10^{-1}	10^1
Limit case I	10^{-5}	10^{-1}	10^1
Limit case II	10^{-1}	10^{-5}	10^3
Limit case III	2×10^{-5}	10^{-5}	10^3

Table 1.5: Values of the fixed parameters in the various cases studied. General case: $\Delta t \ll T_L, \tau_p$. Limit case I: $\tau_p \ll \Delta t \ll T_L$. Limit case II: $T_L \ll \Delta t \ll \tau_p$. Limit case III: $\tau_p, T_L \ll \Delta t$.

1.3 Results

Two-phase flow simulations of system 1.1 have been performed in the various cases previously mentioned. The system considered consists in purely diffusive motion of particles, as seen in Figure 1.2.

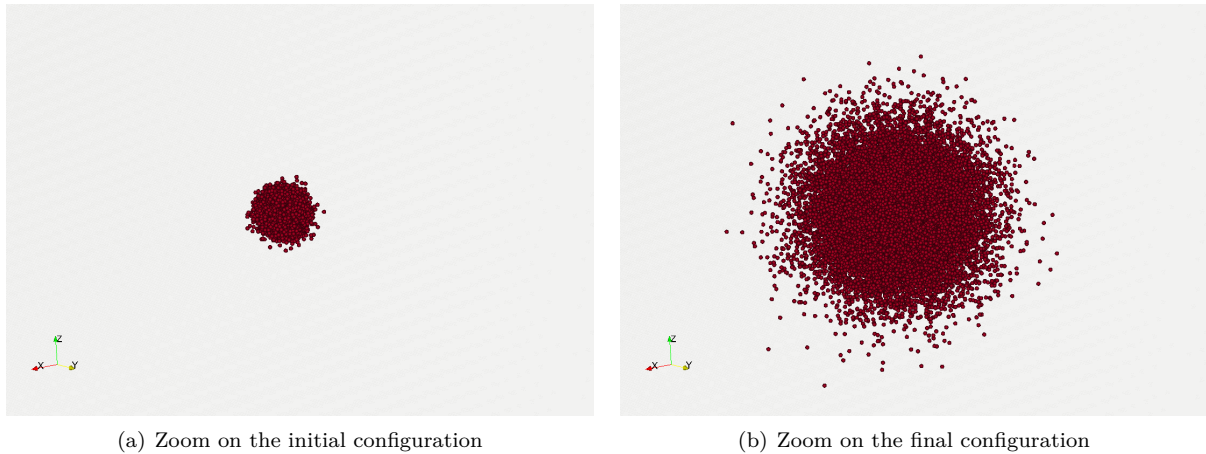


Figure 1.2: 3D view of the particles inside single cell (zoom) at the initial and final time step.

Results are displayed in Figures 1.3-1.6 for the second-order moments (first-order moments are omitted since the solutions of system 1.1 are Gaussian random variables with a mean equal to zero) in the various cases. It can be seen that all numerical results are close to the analytical solution.

EDF R&D	Validation of <i>Code_Saturne</i> 6.0.2 (Appendix 1)	????-????-2019-????-EN Version 1.0
---------	---	---

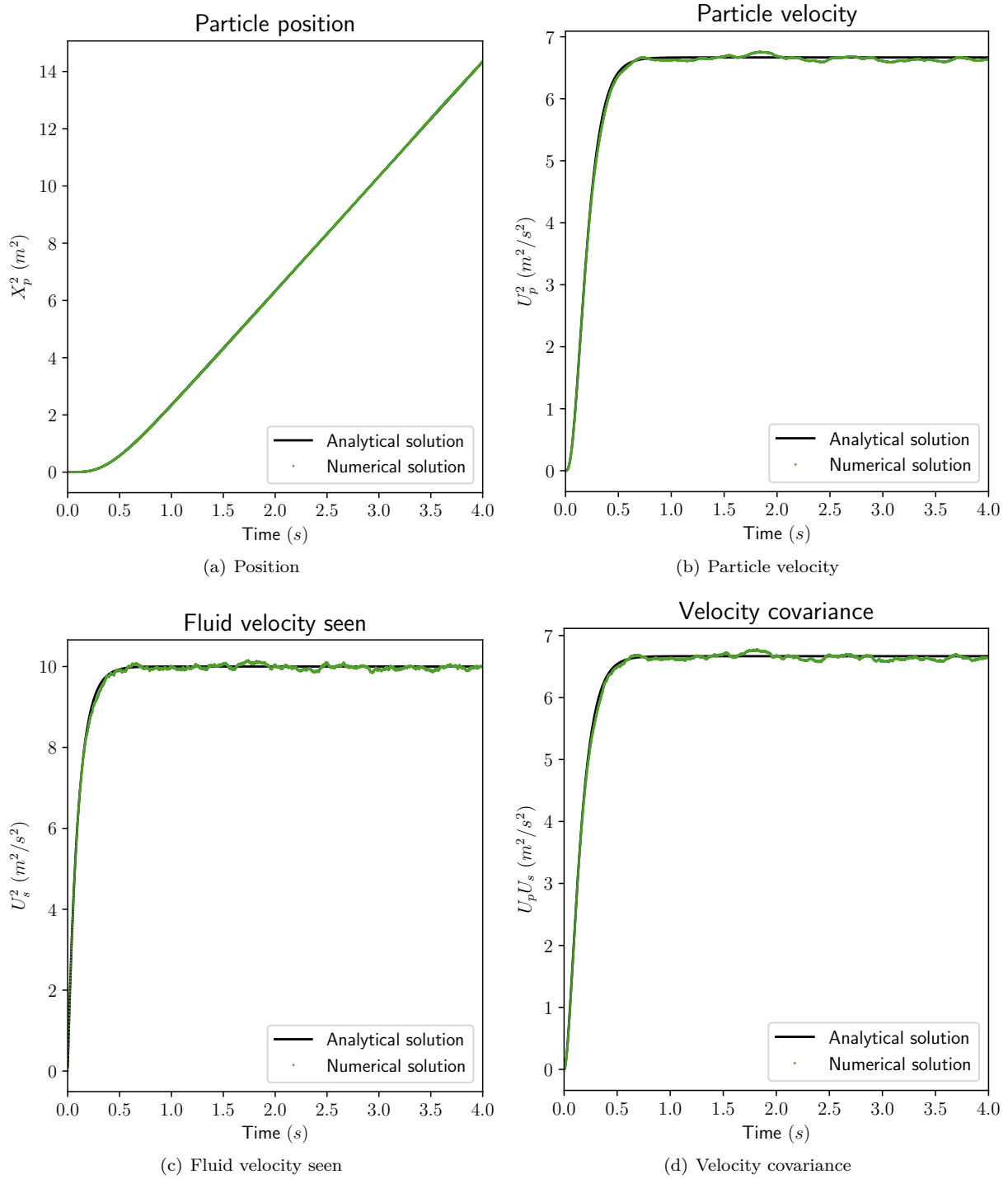


Figure 1.3: Comparison of numerical results for system 1.1 (symbols for 1st and 2nd order schemes) and analytical solutions (lines) in the general case.

EDF R&D	Validation of <i>Code_Saturne</i> 6.0.2 (Appendix 1)	????-????-2019-????-EN Version 1.0
---------	---	---

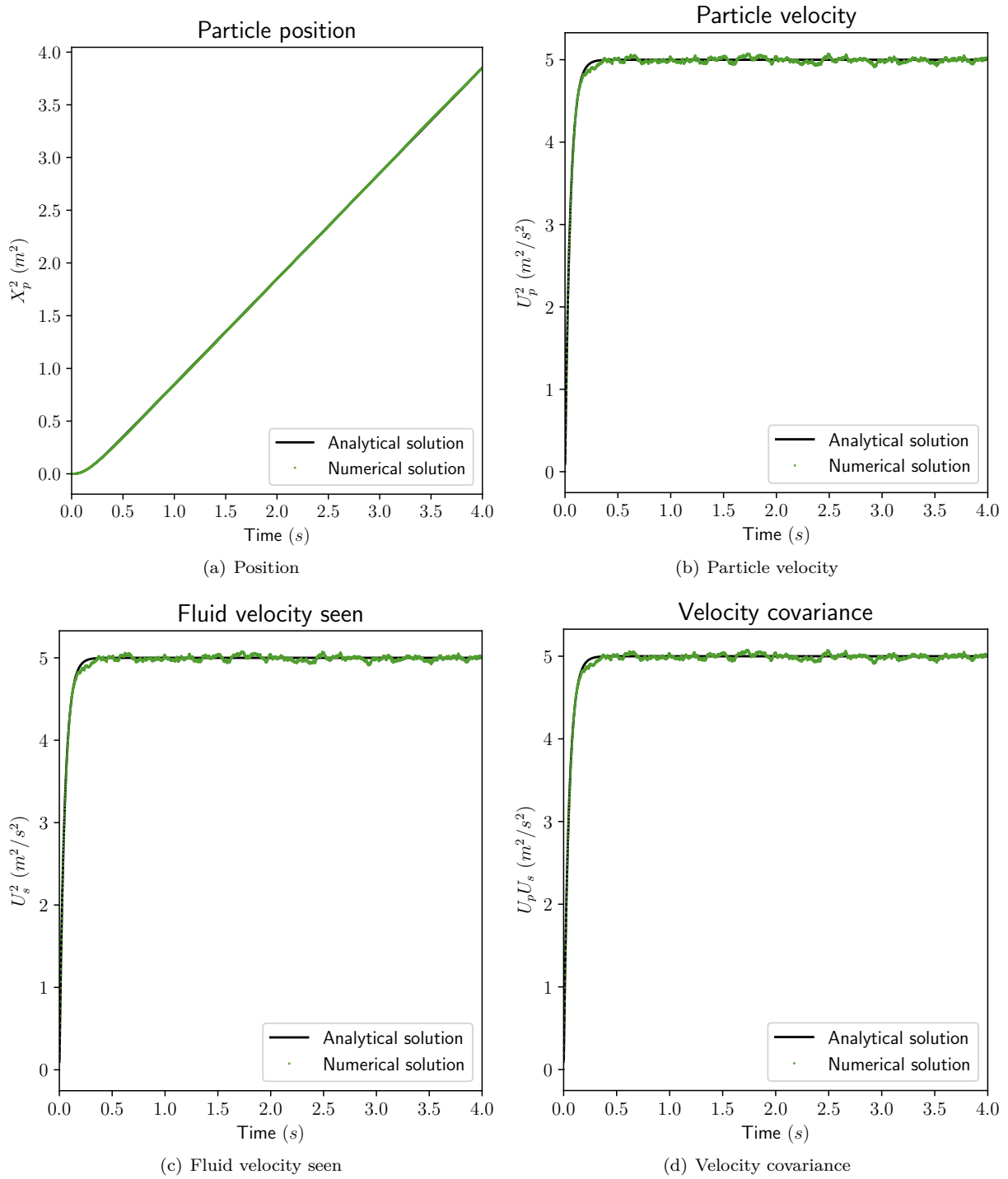


Figure 1.4: Comparison of numerical results for system 1.1 (symbols for 1st and 2nd order schemes) and analytical solutions (lines) in the limit case I.

EDF R&D	Validation of <i>Code_Saturne</i> 6.0.2 (Appendix 1)	????-????-2019-????-EN Version 1.0
---------	---	---

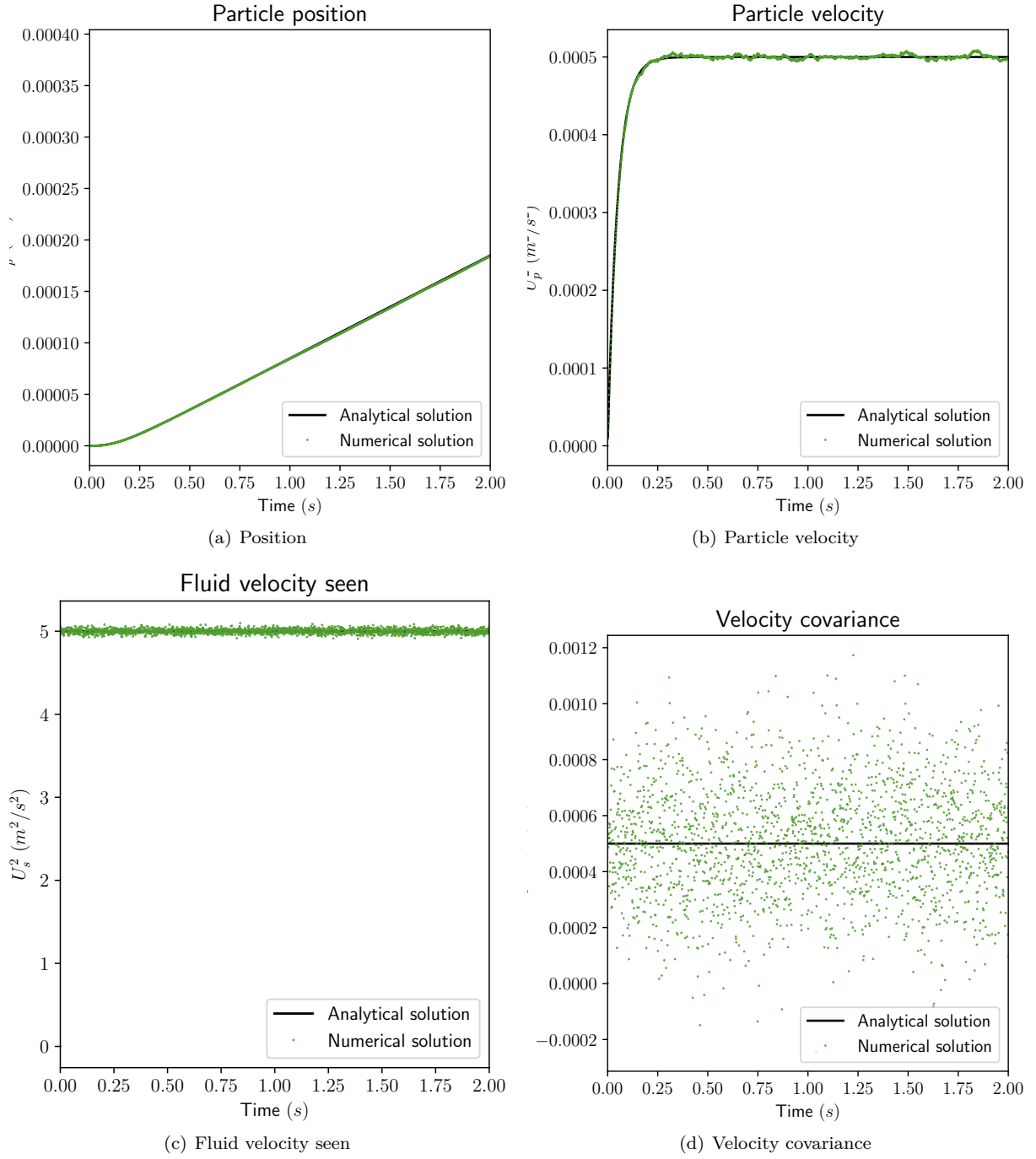


Figure 1.5: Comparison of numerical results for system 1.1 (symbols for 1st and 2nd order schemes) and analytical solutions (lines) in the limit case II.

EDF R&D	Validation of <i>Code_Saturne</i> 6.0.2 (Appendix 1)	????-????-2019-????-EN Version 1.0
---------	---	---

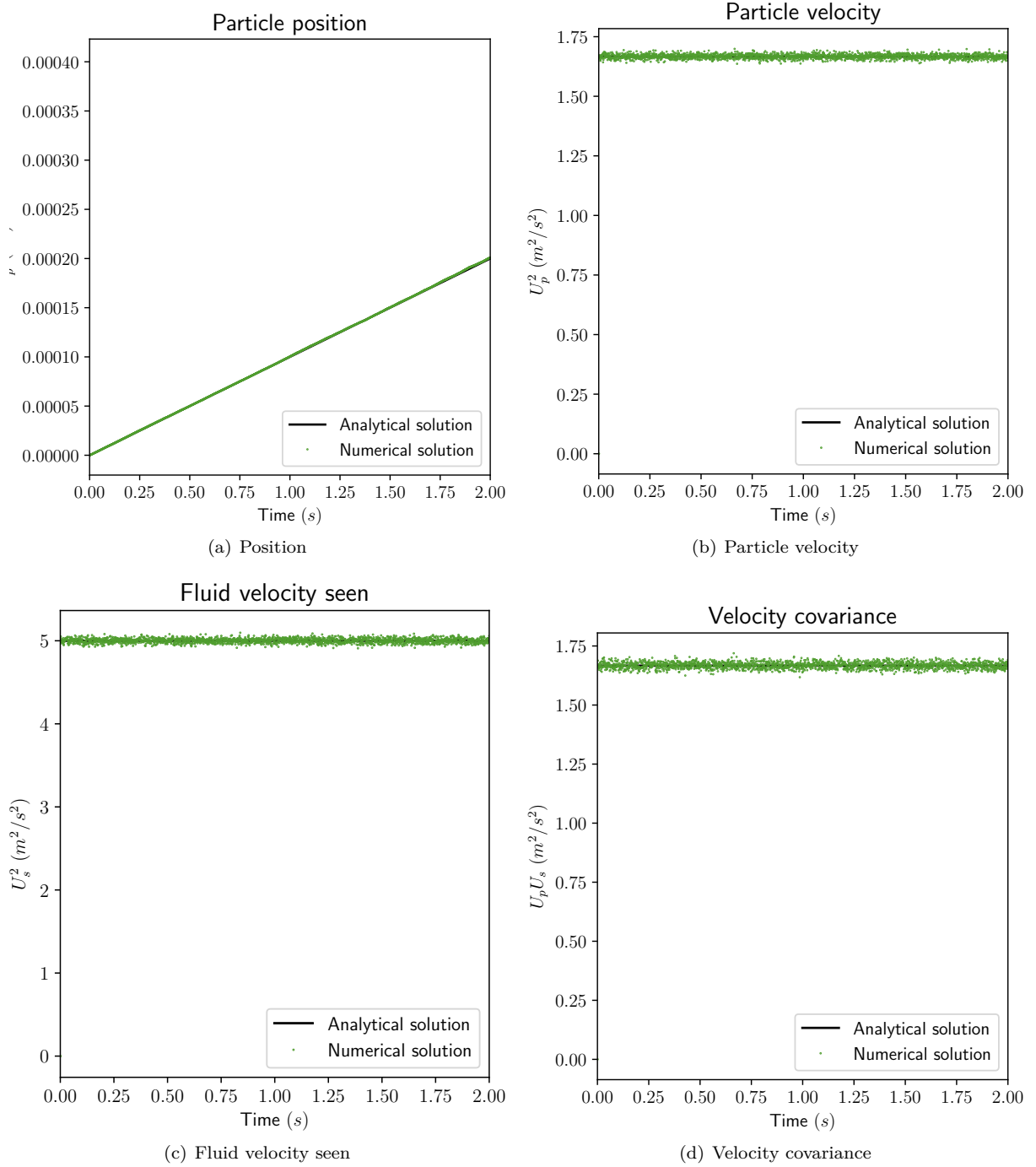


Figure 1.6: Comparison of numerical results for system 1.1 (symbols for 1st and 2nd order schemes) and analytical solutions (lines) in the limit case III.

1.4 Conclusions

Overall, this algorithm for the numerical scheme is precise and robust even in all the ideal cases considered.